

SEC

2

journal

Software Engineering Center

2005年4月25日発行
第1巻 第2号 (通巻2号)
ISSN 1349-8622

SEC Journal No. 2

第1巻第2号 (通巻2号)
2005年4月25日発行 ©独立行政法人 情報処理推進機構

「エンピリカルソフトウェア工学の現状と展望:SELが遺した13の教訓」

松本健一

「ソフトウェア開発の生産性管理に基づく見積りモデル」

太田忠雄

「組込みシステムとユーザビリティ工学」

平沢直毅

編集兼発行人

F113-6591 東京都文京区本駒込2-28-8 文京グリーンコート センターオフィス16階
独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター
所長 鶴保 征城

独立行政法人 情報処理推進機構

Tel:03-5978-7543 Fax:03-5978-7517
URL:http://www.ipa.go.jp/
定価1,470円 (本体1,400円)



IPA
独立行政法人 情報処理推進機構



IPA
独立行政法人 情報処理推進機構
<http://www.ipa.go.jp/>



1	巻頭言 藤原 武平太(独立行政法人 情報処理推進機構 理事長)
2	所長対談：重松 崇 トヨタ自動車株式会社常務取締役 車載用組込みソフトウェアの開発に ソフトウェア工学を取り込む
6	招待論文 エンピリカルソフトウェア工学の現状と展望： SELが遺した13の教訓 松本健一(奈良先端科学技術大学院大学)
14	ソフトウェア開発の生産性管理に基づく見積りモデル 太田忠雄(株式会社ジャステック)
22	組込みシステムとユーザビリティ工学 平沢尚毅(小樽商科大学)
30	海外レポート 鶴保証城
32	技術解説 ソースコードの品質向上技術 川井奈央
38	PMBOKの概要 井沢澄雄
44	ソフトウェア開発の現場から 要求工学への期待 荒生知之
50	組織紹介 社団法人トロン協会トロンプロジェクト 大橋 博
52	北陸先端科学技術大学院大学 e-Societyプロジェクト 片山卓也・岸知二・青木利晃・岡崎光隆
54	慶應義塾大学 天野研究室 天野英晴
56	Project Report 先進ソフトウェア開発プロジェクト 樋口 登
58	SEC成果報告 報告書 / 執筆活動、取材記事
59	Web活動 / 出版情報 / SEC成果物
60	ものづくり競争力共同研究
60	IESEとの共同研究
61	イベント報告
62	BOOK REVIEW
63	ソフトウェア・エンジニアリング関連 イベントカレンダー
64	あとがき
65	お知らせ(SEC Forum 2005 / バックナンバー)

巻頭言

ソフトウェア産業の未来と ソフトウェア・エンジニアリング・センターの役割



独立行政法人 情報処理推進機構
理事長 藤原 武平太

厳しい競争の果てに見えるもの

特定サービス産業動態調査によれば、平成16年の情報サービス業の売上高前年比は0.5%増と、ほぼ横ばいの成長となっていますが、収益を中心とした経営者の実感はそれ以上に厳しいものとなっているのではないかと思います。

その主な原因は、いわゆる不採算案件の増加です。ユーザからの価格引下げ圧力が強まる一方で、見積り甘さや、ユーザとのコミュニケーションの不足、プロジェクトマネジメントの不十分さなどの要因が重なって、収益を大きく悪化させている例が増えています。そのため、ソフトウェア産業全体として従来のような右肩上がりの成長が期待しにくく、中国、インドとの競争が激しくなってくる環境では、このような不採算案件が経営を揺るがすという状況が深刻化すると考えざるを得ません。

飛躍のチャンスととらえる

しかし、企業の合併に伴う案件、ユビキタス的な案件、これまでよりもはるかに厳しい品質・納期を持つ案件等、従来型ではない案件が増えてきているのも事実です。これら新しい領域の案件は、ソフトウェア産業界にとっては魅力的な収益源に成り得ますが、従来のソフトウェア・エンジニアリングの方法論での対応は、ビジネスモデルも含めて難しいものともいえます。今こそ、新領域に対応した新しいソフトウェア・エンジニアリングを考

える非常によいチャンスだといえるのではないのでしょうか。

SECの役割

このチャンスを生かし、ソフトウェア産業界のスキルの底上げをしようというのが、私共IPAの中に、昨年10月設立したソフトウェア・エンジニアリング・センター(SEC)です。上述した課題は、これまで個々の企業の経営に委ねられ、オール・ジャパンとしての焦点が当たっていない分野でした。数字に基づく精密な経営を行うための基礎を構築し、従来の必ずしも合理的でない慣行を打破して合理的なビジネスモデルを定着させていくためには、ユーザ、ベンダー双方の業界の協力、関連する学界の英知、そして、政府の政策的支援が不可欠であります。SECは、産学官の協力を得て、このような課題に取り組む我が国のセンターとして活動していきます。

ソフトウェア産業界の方々には、SECが取り組んでいく課題は、ソフトウェア・エンジニアリングという分野の専門家に委ねておけばよい問題ではなく、経営者が経営そのものの課題として、一緒になって取り組んでいくべきであることを、是非ご理解いただきたいのです。SECは生まれたばかりの組織ですが、ソフトウェア産業の健全な発展のために全力を挙げて取り組んでいきますので、どうか、私共の活動に参加し、成果を活用していただきたいと考えています。

車載用組込みソフトウェアの開発にソフトウェア工学を取り込む

国際競争力が重視される中、いま日本におけるものづくりの価値が再認識されつつある。とくに、わが国が誇る電子機器などのシステムにおいてはなおさらである。こうしたシステムの心臓部を司る重要なキーテクノロジーの1つが「組込みソフトウェア」である。マイクロコンピュータなどに代表される組込みソフトウェアは、先の経済産業省による2004年版組込みソフトウェア産業実態調査によっても、市場規模2兆円、ソフトウェア技術者数も約15万人に達し、益々、大容量化、高機能化、複雑化してきた。それだけに、今後の開発力アップのためにはさらなる人材増強のほか、さまざまな対策が望まれる。ここでは、その辺りの実情と課題、展望を探るべく車載用組込みソフトウェアにフォーカスした。



鶴保 征城(つるほ せいしろう)
1966年大阪大学大学院工学研究科電子工学専攻修士課程修了後、同年4月日本電信電話公社(現NTT)入社。1989年11月NTTソフトウェア研究所長、NTTデータ通信株式会社取締役開発本部長、同社常務取締役技術開発本部長、NTTソフトウェア株式会社代表取締役社長を歴任し、2004年6月独立行政法人情報処理推進機構 参与、同年10月ソフトウェア・エンジニアリング・センター所長に就任。高知工科大学工学部情報システム工学科教授(2003年~) 奈良先端科学技術大学院大学 客員教授(2003年~) 独立行政法人 日本学術振興会「基盤的ソフトウェア技術開拓」に関する研究開発専門委員会委員(2003年~) 社団法人 情報処理学会 会長(2001年~2002年) XMLコンソーシアム 会長(2001年~) 社団法人 電気情報通信学会 フェロー 社団法人 情報処理学会 フェロー

鶴保 トヨタ自動車では、たくさんのマイコンが使われていますが、いつごろから使われはじめたのでしょうか。

重松 自動車の制御に組込み用マイコンが使われるようになったのは25年ほど前です。このときは排気ガス規制に対応するためのエンジン制御向けが主で、エンジンサイクルにおける燃料噴射やその量、点火などが制御されていました。その後、排気ガス規制は一段と厳しさを増し、エンジンサイクルのタイミングが、数μ秒のずれさえも許されなくなりました。また、VSCと呼ぶスピン防止システムは、たとえば高速走行時に滑りそうになると自動的に制動をかけるというものですが、このときマイコンが路面状態や走行状態などを推定しながら制御します。1つ間違えると大事故につながりますので、ここでも極めて高い信頼性が要求されました。人命を預るソフトウェアですから、慎重に作っています。

車載用組込みソフト開発の実態

鶴保 車載用は、組込みの中でも実時間性等最も厳しい要求が求められる領域の1つに数えられると思います。プリウス等多くのマイコンを搭載される車種も増えてきていますがこれからの開発に必要な課題をお聞かせください。

重松 とくに大きな課題は、車載用組込みソフトウェアの風土そのものをどう変えるか、ではないでしょうか。メカ制御からチップベースのマイコン制御に移行する際に、ソフトウェア開発手法や技法をほとんど意識せずに今日に至ってしまい、ドキュメントを残すという仕組みの定着を待たずに、技術進歩自体が早かった。また、メカのエンジンを知らずしてエンジン制御のソフトウェアなんて書けませんが、両方のわかる人を育てる余裕がなく人材も不足してしまいました。そのため、これまではサプライヤとの“すり合わせ”で作ってきました。それでも品質面から見ると日本は欧米と比較して10倍以上良いですし、開発期間や柔軟性も数倍以上勝っています。ですが、欧米は、ドキュメントをきちんと残しています。

鶴保 ドキュメントを残さないことには何か理由があるのでしょうか。

重松 自動車では制御対象自体がある程度ロバスト性をもっていますし、制御の理想状態に対してメカ的な遅れを補償する等のノウハウのような部分があります。そのためサプライヤが自動車メーカーとの“すり合わせ”を行い、短期間でまとめ上げていくので、途中過程でドキュメントを作らずにもの作りに入ってしまう。

鶴保 すり合わせの暗黙知を形式知に変えていくためにも、これからは日本でも欧米のように、構造設計をしたりドキュメントを残すほうがよいと思います。

重松 そうなるよう、いま我々が働きかけをしているところです。これからは、モジュール化や構造化に取り組まないと、とても人材不足に追いつけません。かといって、これだけをメインにしていようものなら、まず欧米には負けてしまうでしょうね。日本の強みである“すり合わせ”の方法を生かしながら、構造化をうまく使っていくといった、日本固有の組込みソフトウェア技術がベストでしょう。しかしその一方で“すり合わせ”ではなく、汎用化できそうなところはいろいろなものを組み合わせるという方法も必要です。こうした使い分けをするためには、仕様を分析し要件を整理して正確に記述することによって重要なノウハウのあるところと、ないところに分ける方法が肝要です。こうした方法でまさに現場を変えようとしているのです。

鶴保 ITに対する資本投下が企業の生産性向上をもたらすためには、それ以外の目に見えない要素が重要だと言われています。しかし、これはITに対する資本投下の重要性を否定するものではありません。このことを今のお話にあてはめると、構造化やプロジェクトマネジメント以外の要因も重要ですが、これらを統合すれば、効果はきっとあるはずということでしょう。

重松 ドキュメントを書いていない部署に対して「書きなさい」というと「なぜなんだ!」という反発があって苦慮するところだと思いますが、しかし書き始めてドキュメントが見えてくると、多くの人たちによるDesign Reviewができて、品質も上がっていくはずで、Design Reviewは大変おもしろくて、日本が強みとする“すり合わせ”のキーになっているかもしれません。ソフト側の言い分は「要求書をきちんと書いてくれ」ですが、システム側は「今まで一緒にやってきたのだからそれくらいは推測して作ればいいじゃないか」です。このギャップをDesign Reviewが埋めます。

トヨタ自動車における車載用組込みソフトウェア開発のスタンス

鶴保 トヨタ自動車自体が作成する組込みソフトウェアにはどのようなものがありますか。

重松 自動車の制御システム開発の中では、メカ部品



重松 崇(しげまつ たかし)
1975年熊本大学大学院工学研究科修士課程機械工学専攻修了。同年トヨタ自動車工業株式会社(現トヨタ自動車株式会社)に入社。2001年同第5開発センター第2電子技術部長、2004年同常務役員就任。株式会社トヨタマップマスター取締役、株式会社トヨタIT開発センター取締役を兼務。

の機能を電子制御・ソフトで補間する意味合いの部分が大きく、システムエンジニアが実機を動かしながらソフトを変更して機能を確認する方法をとります。その場合、ドキュメントで表すよりも、実際に使用したソフトをサプライヤに渡して意思疎通を図ることになります。

鶴保 最終的には、トヨタ自動車を作ったものをサプライヤに渡すと、彼らが部品に組み込んで納品するといった形態ですね。

重松 トヨタ自動車から渡すソフトには構造化や再利用を考慮されたものではない為、サプライヤ側では受け取ったソフトから仕様を解釈し、構造化を考慮して作り直すこととなります。その為、最終的にはサプライヤで作成したソフトでの実機動作再検証も必要となり、場合によっては仕様の誤解釈により設計・検証の手戻りが大きな問題になってしまいます。また、エンジン制御は自分たちで行いますが、ナビゲーションやボディ等は、ほとんどをお任せしています。

鶴保 しかし、各システムはネットワーク(CAN⁽¹⁾)で接続されるのですね。

重松 実はそこが問題なのです。サプライヤの数が多いので、品質不足のサプライヤが1社でも存在すると、CANがうまく機能してくれません。そこでどうサプライヤそれぞれのレベル合わせを行うかがポイントになります。つまり品質監査的なことを行って、品質が悪ければ

そのサプライヤにはソフトウェアを作らせないので。そこで、品質のよいサプライヤから購入したたとえばCANのソフトウェアを、トヨタ自動車から支給部品としてそのサプライヤに支給します。当然支給されるサプライヤは、中味がわからないにもかかわらず組み合わせる責任は持たされますので不満が残りますし、その責任の取り具合もきわめて難しいものがありますが。

新たなビジネスの可能性

鶴保 金型メーカーは先行して設計する、とよく言われますが、ソフトウェアでもサプライヤが先行するような傾向は見られますか。

重松 それはありますが、キーになるところはトヨタ自動車が行います。

鶴保 ソフトウェアのみの受発注はあるのでしょうか。

重松 エンジン制御はそうですね。ボディ系も最近少しはそうなりつつあります。ECU⁽²⁾等ハードウェアは単機能ごとに発注していましたが、それだと高価になりますので、この機能とこの機能を一緒にしたECUを作る、といったことを行います。ハードウェアの再構築を行おうというわけです。こうなると、あるサプライヤ側のソフトを、作らなくてもいいといわれたサプライヤ側に売ることになります。これは新しい形態のビジネスモデルといえます。

鶴保 難しい流れと思いますが、それはサプライヤ同士で行われますか、それともトヨタ自動車が仲介しますか。

重松 トヨタ自動車が入らないと流れが悪くなりますので入らないと難しいでしょうね。

組み込みソフトウェア作成のための教育体制

鶴保 ところで、教育の問題ですが、先頃トヨタでは大学も創設されましたが、今、教育にはどのような取組みを展開しておられますか。

重松 自分でC言語によりソフトウェアを組んで、デバッグしてといった程度のルーチンであれば人材はいます。しかしソフトウェアの構造設計ができる人、つまりアーキテクトをどう育成すべきかに悩んでいます。この辺りの教育がうまくいく方法を見いだすことは重要です。

鶴保 そうしたソフトウェアに関して、海外との連携はいかがですか。

重松 実際に中国とは数百人オーダで取り組んでいます。インドにあるヨーロッパ系サプライヤでは、約700人中自国の人が2~3人で残りはインドの人たちでした。彼らがソフトウェア開発を行っているのです。

鶴保 それは車の部品ですか。

重松 ボディ系がメインで、制御系エンジンは部品のみでした。ここでの教育はうまくいっていて、本車で2年間教育の後、現場につかせるという形です。

鶴保 実は、エンタプライズの方もアウトソーシングが考えられていますが、業務仕様がなかなか確定しないという状況もあり、まだこれからということですね。この面では、組込みの方が、アウトソーシング化しやすいとも考えられますね。

重松 確かにそれはいえます。しかし、もし、現状のように現場が悲鳴をあげた状況でアウトソーシングを行うと、それこそ海外に丸投げという悪いパターンになりかねません。そうするとノウハウが流れてしまう不安があります。

鶴保 ということは、ローコストで技術力がある会社に対しても、トヨタ自動車としては、海外へのアウトソーシングには慎重だということですね。

重松 アウトソーシングする場合は、ちゃんと切り出した形で出すことが必要でしょう。つまり、最初は下流域でデバッグ作業を切り出して、中流工程ではツールでノウハウ流出を止めてしまうといったようなことです。

車載用組み込みソフトウェアの標準化の行方

鶴保 今後、車は多機能化の一途を辿ることになるでしょうね。そうすると、これまでの車では見られなかったようなノウハウが必要になってくると思いますが。

重松 最大の問題点は、エンジンとブレーキとボディをリンクさせて統合制御することです。ハイブリッドカー技術はエンジンとミッションとブレーキを同時に制御しています。エネルギーを回生するためにモータでブレーキをかけて、バッテリーを通り、ここでフル充電されるとブレーキをかけるといったようなものです。これがLANにより接続され動作しますからなかなか難しい技術で、サプライヤサイドでは難しいでしょう。この統合

システムは、自動車メーカーがおさえなければなりません。

鶴保 半導体大手のインテルでは、何百人ものコンピュータアーキテクトがいます。彼らは、自分たちのチップを売るためにその使い方を研究しているのです。御社でもソフトウェアのアーキテクチャや機能の使い方など、相当広範な技術者集団を保有されるのではないのでしょうか。

重松 そうですね。システム屋がソフトウェアのことをちゃんと知っていて、ヒューマンインタフェースやLAN、CPUのコアなどを理解している、こうした意味でマルチ的な人材をたくさん抱えているといいですね。たとえばJASPAR⁽³⁾という組織では、いまFlex Ray⁽⁴⁾中心の議論を展開している段階ですが、自動車に馴染みのよいチップとはどのような構造なのかも追求していきます。

鶴保 サプライヤがヨーロッパへ輸出する場合などは、ヨーロッパのアーキテクチャの範囲での部品調達といった考え方が出てくるのではないのでしょうか。エンタプライズの方でも、アメリカ的なデファクトが要求されます。

重松 JASPARの任務にAUTOSAR⁽⁵⁾と連携するミッションがありますが、JASPARからAUTOSARを通じてAUTOSARのインプリメントを補完していきます。ここで積極的に日本発の提案をしたいですね。将来的にはサプライヤが国内と海外向けに2種類取り組むことがなくなるようにしたいと思います。

鶴保 かつて通信業界では、国内ベンダがNTT向けと海外向けの2本立てで取り組んでいました。自動車業界は通信業界を凌駕するようになりつつありますから、舵取りはさらに難しいでしょうね。

重松 そうです、AUTOSARなどは自分たちのものを確立させたいと一生懸命ですから、ネゴシエーションをうまくして、本当にいいものを提案しなければなりません。

鶴保 携帯電話の組み込みソフトウェアは5~6Mバイトに膨らんでいますが、この原因の1つは、GSM、W-CDMA、cdma2000といった多種多様なプロトコルを搭載させるようになったからです。標準化は技術の進歩という面から見ると、必ずしも好ましいことではありません。



それならば、むしろそれぞれの方式で技術をどんどん進めた方がいい。肥大化のしわ寄せはすべてソフトにくることになります。車の場合でも、サプライヤが大変なことになるのでしょうかね。

重松 しかし、逆も考えられます。もしトヨタ自動車が欧米のサプライヤを使う場合、しかも閉じたシステムではなくCANによりネットワーク接続されたシステムの場合、彼らがこちら側に揃えなくてはなりません。したがって、自動車メーカーとしては、極力、仕様を揃え得るところは揃えたいという気持です。

先進ソフトウェア開発プロジェクトへの取り組み

重松 トヨタ自動車では、SECの先進ソフトウェア開発プロジェクトにはITセンタで取り組む予定です。インフラ共通部分のデータのインタフェースの仕様策定では歩調を合わせ、インフラで取り上げたデータを元に加工することはうまくいくでしょう。メーカーの独自性は、アプリケーションで出していくことになるかと思っています。

鶴保 そうですね、インタフェースを協議して共通化させていくことは可能でしょう。

重松 いかにか大手自動車メーカーであっても、各メーカーが独自に超高速多重通信用チップを作ろうとしてもコスト的に見合わないはずですね。したがって各社でハードウェアを揃えようということになり、いまJASPARでも半導体メーカーへ標準的な要求仕様が出せるように取り組んでいます。

鶴保 ぜひ車における組み込みソフトウェア一層の発展のためにご尽力されますことを期待しております。SECといたしましても、今後ともさまざまな働きかけを業界に向けて行っていきます。

(1) CAN : 自動車内の高速通信ネットワーク規格

(2) ECU : Electronic Control Unit

(3) 有限責任中間法人JASPAR : Japan Automotive Software Platform and Architecture , <http://www.marklines.com/ja/jaspar.jsp>

(4) Flex Ray : 車両用LANのプロトコル

(5) AUTOSAR : The Automotive Open System Architecture , <http://www.autosar.org/>

エンピカルソフトウェア工学の現状と展望: SELが遺した13の教訓

Beyond the SEL: 13 Lessons Learned by SEL and EASE/SEC Approach in Empirical Software Engineering



奈良先端科学技術大学院大学教授
工学博士
松本 健一

Nara Institute of Science and Technology, Professor, Dr. Eng.
Kenichi Matsumoto

ソフトウェアやその開発過程から得られる定量的データに基づいてソフトウェアの生産性や品質の向上を目指す実証的アプローチ(エンピカルアプローチ)が注目されている。本稿では、米国SEL (Software Engineering Laboratory) の研究者たちが遺した「プロセス改善におけるエンピカルアプローチに関する13の教訓」を紹介すると共に、それら教訓と対比させる形でEASE (Empirical Approach to Software Engineering) プロジェクトの現状とSECへの期待について述べる。

This paper shows 13 lessons learned about the empirical approach to software process improvement by members of the Software Engineering Laboratory, in order to clarify the current status of the EASE project which aims to establish Empirical Software Engineering to improve software productivity and reliability in Japan. This paper also shows some expectations for SEC about sharing empirical data of software projects among software industry members.

Key Words & Phrases: エンピカルソフトウェア工学, ソフトウェア測定, ソフトウェアメトリクス, ソフトウェアプロセス改善
Empirical Software Engineering, Software measurement, Software metrics, Software Process Improvement

1 はじめに

近年、ソフトウェアやその開発過程から得られる定量的データに基づいてソフトウェアの生産性や品質の向上を目指す実証的アプローチ(エンピカルアプローチ)が注目されている[1]。受注ソフトウェア開発を対象として社団法人情報サービス産業協会が行ったアンケート調査の結果[2]でも、ソースコード行数、ファンクションポイントなど規模に関して何らかの測定を行っている企業は85%、不具合数/密度、レビュー回数、テストケース数など品質に関して何らかの測定を行っている企業は93%にもものぼる。

ただし、(苦労して)測定したデータの利用率は必ずしも高くない。同調査結果によれば、コスト見積りを測定データに基づいて行っている企業は51%に過ぎず、43%の企業では依然として、担当者の経験や類推に基づく見積りが行われている。エンピカルアプローチをソフトウェア開発に定着させるためにはまだまだ工夫と努力が

必要なのである。

本稿では、米国SEL (Software Engineering Laboratory) の研究者たちが遺した「プロセス改善におけるエンピカルアプローチに関する13の教訓」を紹介する。次に、著者らが2003年度から推進しているEASE (Empirical Approach to Software Engineering) プロジェクトの現状を、それら13の教訓と対比させながら述べる。最後に、SECが取り得る、SELよりも高次のエンピカルアプローチの可能性について指摘する。

2 SELが遺した教訓

SELは、ソフトウェアプロセス改善の研究を行うために、University of Maryland (UMD) のV. R. Basili教授らによって1976年に設立された組織である。その具体的な研究目標は次のとおりである。

- NASA/Goddard Space Flight Center (NASA/GSFC) におけるソフトウェア開発プロセスの理解
- 様々な方法論, ツール, モデルが持つ影響力の測定

・ソフトウェア開発の成功例の収集と実際のソフトウェア開発への適用

2001年にその25年の歴史を閉じるまでの間、NASA/GSFC向けソフトウェアの開発において測定したデータに基づいて、QIP (Quality Improvement Paradigm), GQM (Goal/Question/Metric), EF (Experience Factory) といったフレームワークを提案するとともに、250編を超える学術論文を発表している。その功績が認められ、1994年には、米国IEEE (The Institute of Electrical and Electronics Engineers) とSEI (Software Engineering Institute) によるProcess Achievement Awardの最初の受賞者となっている。

ただし、研究の様々な局面で十分に取り組むことのできなかった課題があったことをSELのメンバーは認めている。彼らはそうした課題を「プロセス改善におけるエンピカルアプローチに関する13の教訓」(図1)としてまとめている[3]。教訓は次のとおり4つのカテゴリーに分類されている。なお各教訓につけられた番号はSELにおいて得られた順番を示す。

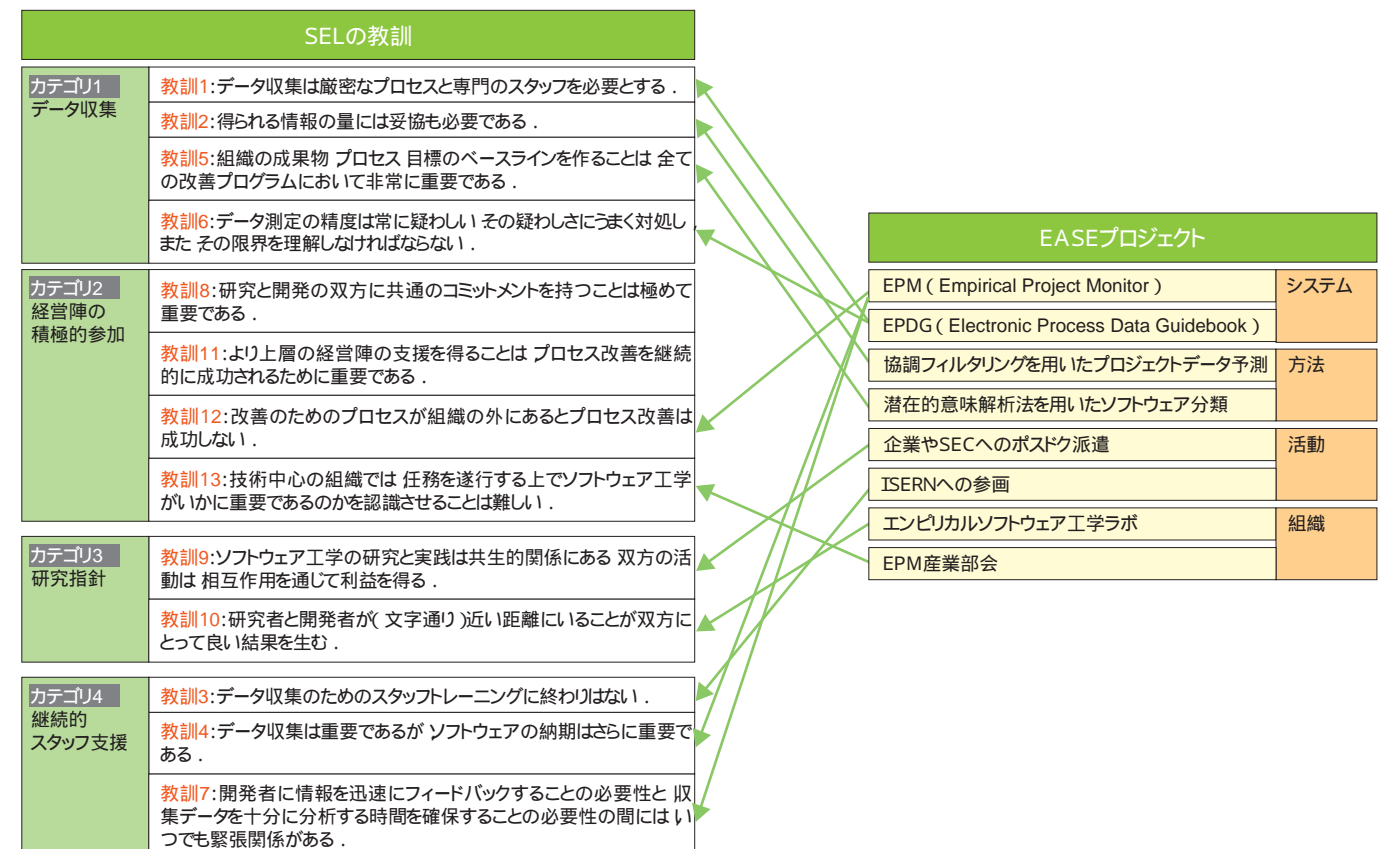


図1 SELの教訓とEASEプロジェクト

カテゴリ3 研究指針に関する教訓

教訓9：ソフトウェア工学の研究と実践は共生的関係にある。双方の活動は、相互作用を通じて利益を得る。

教訓10：研究者と開発者が（文字通り）近い距離にいることが双方にとってよい結果を生む。

カテゴリ4 継続的スタッフ支援に関する教訓

教訓3：データ収集のためのスタッフトレーニングに終わりはしない。

教訓4：データ収集は重要であるが、ソフトウェアの納期はさらに重要である。

教訓7：開発者に情報を迅速にフィードバックすることの必要性和、収集データを十分に分析する時間を確保することの必要性の間には、いつでも緊張関係がある。

3 教訓を踏まえたエンピカルアプローチ

3.1 EASEプロジェクト

EASEプロジェクトは、文部科学省リーディングプロジェクト「e-Society基盤ソフトウェアの総合開発」の一環として2003年から5年計画で実施されている[4]。その目標は、エンピカルソフトウェア工学の成果をソフトウェアプロダクトとして具体化し、産学官連携のもとに、エンピカルアプローチの普及と更なる発展を実現することにある。

プロジェクトの詳細については、プロジェクトホームページ[2]等をご参照いただくとして、ここでは、先に示した「SELが遺した13の教訓」との対応関係を示しながら、開発中の主なシステムやデータ分析方法、実施中の活動や組織について述べる（図1）。

3.2 システム

(1) EPM

EPM (Empirical Project Monitor) は、現在広く普及している開発支援フリーウェア (CVS, Mailman, GNATS等) と連携することによって、ソフトウェア開発プロジェクトデータをリアルタイムに収集し、定量的データ分析を可能にするシステムである（図2）[6]。EPMを利用したソフトウェア開発では、プロジェクト管理者や開発者は、プロジェクトの進捗状況や作業状況を客観的に把握することが可能となる。プロジェクトの問題点を迅速に見

することで効果的なプロセス改善も期待される。EPMの適用実験では、開発者に大きな作業負荷を与えることなく、対象プロジェクトの状況を分析可能であることが確認されている。

EPMは、構成管理、メーリングリスト管理、障害管理等、広く普及し利用されているソフトウェア開発支援システムからデータを収集する。プロジェクト管理者や開発者がデータ収集のために特別な作業を行う必要はない。データ収集が優先され納期が犠牲になるということではなく「教訓4」を踏まえたシステムとなっている。

また、EPMは導入コストが小さく、開発の環境やスタイルを大きく変更する必要もない。データ収集のためにフリーウェアを導入することに抵抗のある企業もあるであろうが、特定の企業のツールやシステムを導入することに比べれば、導入可能な企業やプロジェクトの範囲は広い。もし自社製開発管理システムなどが既に導入されているのであれば、無理にフリーウェアを導入せず、収集データの形式をEPMの標準データフォーマットに変換するなどに対応してもよい。いずれにしても、プロセスを組織内におくことを容易にする、「教訓12」を踏まえたシステムといえる。

さらに、EPMは、探索的で多くの時間を要するデータ分析作業を支援するため、システム上でユーザが実行したデータ分析の内容を保存、再実行する機能を有している。データのリアルタイム収集機能と合わせて、分析結果の迅速なフィードバックを目指すものであり、「教訓7」を踏まえたシステムといえる。

(2) EPDG

EPDG (Electronic Process Data Guidebook) は、定量的データに基づくプロセス管理の実施を支援するためのシ



図2 EPMにおける分析結果表示例

ステムである（図3）[7]。「測定情報モデル[8]」に基づいた諸測定量の定義や実測値を一元管理することができる。データ収集を行う者には、具体的な分析手順や測定情報モデルの構造、収集手順等についての正しい理解を与える。また、データ分析を行う者には、測定値の持つ意味と用途に対する正確な情報を提供する。

より具体的には、CMMI連続表現[9]における「プロジェクト管理のためのプロセス領域QPM（定量的プロジェクト管理）」と「QPMを支援するプロセス領域MA（測定と分析）」において要求される実践項目（プラクティス）の実現を想定している。現バージョンのEPDGを導入し積極的に活用すれば、MAにおいては「CMMIの連続表現における能力レベル3」までの実践がすべて達成可能である。また、EPMなど、ソフトウェアプロセスの測定や分析結果のフィードバックが可能なシステムと組み合わせることで、QPMとMA双方の領域において、より高い能力レベルの達成を支援可能である。CMMIに沿った測定が行える点からも、EPDGは、「教訓1」を踏まえたシステムといえる。

EPDGは、特定の測定環境への依存を避けるため、自動的な測定や分析の機能は備えていないが、過去の測定データや具体的な分析事例を参照することができる。測

定データの精度を直接保証することは難しいものの、測定プロセスの検証や再利用を通じて、測定データの精度向上に間接的に貢献することは可能であり、「教訓6」を踏まえたシステムでもある。

3.3 方法

(1) 協調フィルタリングによるプロジェクトデータ予測

協調フィルタリングはamazon.comの書籍推薦システムなどで用いられている技術で、膨大な情報の中から有益と思われる情報を、システム利用者ごとに選び出すことができる。書籍推薦システムでは、各利用者の未購入書籍それぞれに対して、その利用者が購入した場合に付けるであろう評価点を予測し、評価点の高い書籍の購入を薦める。予測において、他の利用者が書籍に付けた評価点を利用するが、個々の利用者が評価点を付けている書籍は、システムに登録されている書籍のごく一部に過ぎない。いわば、利用できるデータ（評価点）の大部分が欠損した状況（未記入の状態）で予測を行っていることになる。

ソフトウェア開発におけるデータ収集では、データ欠損はある程度避けられない。完了した工程のデータを収集することはできないし、収集システムの一時的な不具

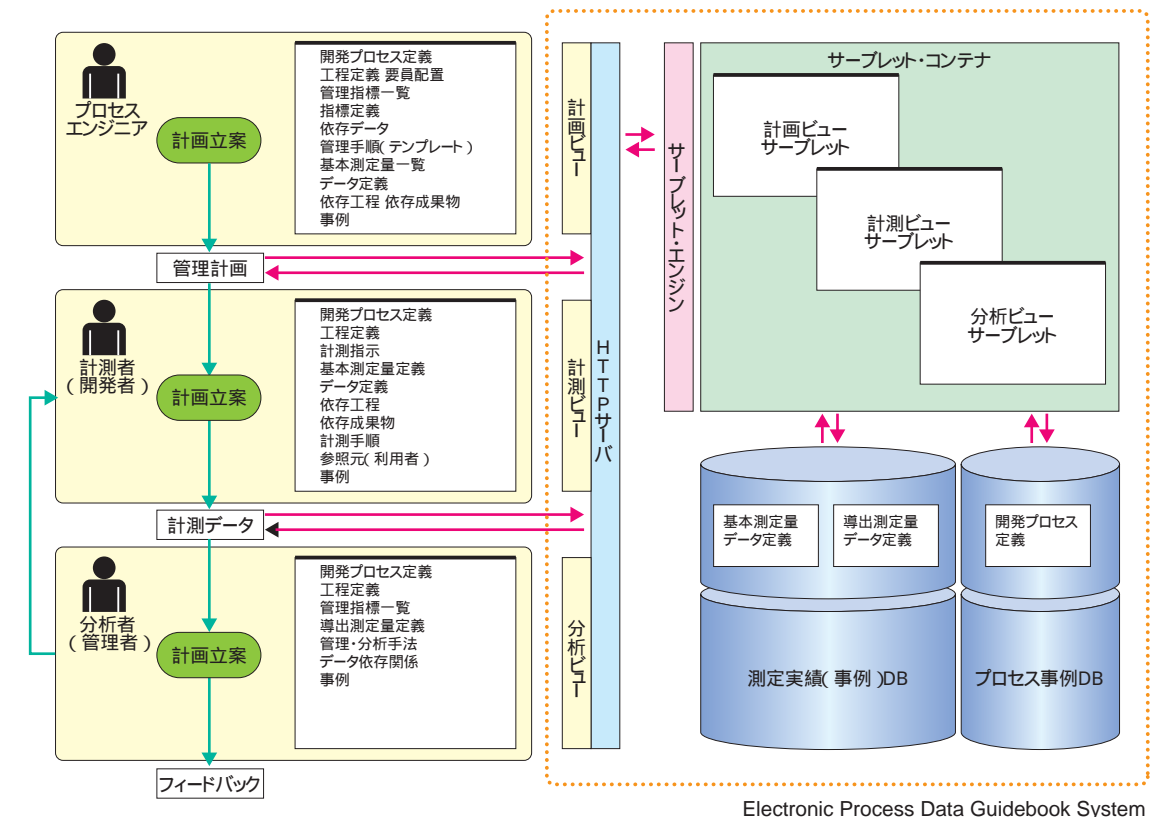


図3 EPDGシステムアーキテクチャ

合などで、ある期間のデータが抜け落ちることもある。また、組織横断的にデータを利用しようとしても、データ分析の目的や開発組織が異なれば、収集データも異なる。テスト工程を評価しようとしているプロジェクトにおいて（評価に直接必要のない）設計工程のデータまで収集する時間的、費用的余裕があるとは限らない。収集データを共通化するために、開発モデル、ツール、ドキュメントなどを全組織で共通とすることも現実的ではない。

重回帰分析やニューラルネット等の多変数モデルでは、ソフトウェアやその開発過程の特性値をできるだけ多く説明変数の候補とすることで、バグ数や開発工数といった目的変数の予測精度を高めようとする。しかし、候補とする特性値が増えれば増えるほど、それらの算出に必要なデータが全て収集されているソフトウェア開発プロジェクト数（データセット数）は少なくなる。データセット数が少なくなると、説明変数が多くてもモデルの予測精度は低くなり、適用範囲も限定される。データ欠損を補う方法も開発されているが、欠損率が30%を超えると予測精度は著しく低下する[10]。

協調フィルタリング技術を用いた予測法では、データ欠損率が70%程度までであれば、バグ数や開発工数などを比較的高い精度で予測することができる。ある企業の1,081件のソフトウェアプロジェクトで測定された14個の特性値（データ欠損率約60%）を用いて試験工数を予測したケーススタディでは、予測値の平均相対誤差やPred25の値が、従来方法（欠損値補完法を併用したステップワイズ重回帰分析）より大幅に改善されることが確認された（図4）[11]。データ欠損を許容する本手法は、得られる情報量の現実的な妥協点を見出しやすくするものであり、「教訓2」を踏まえた分析方法となっている。

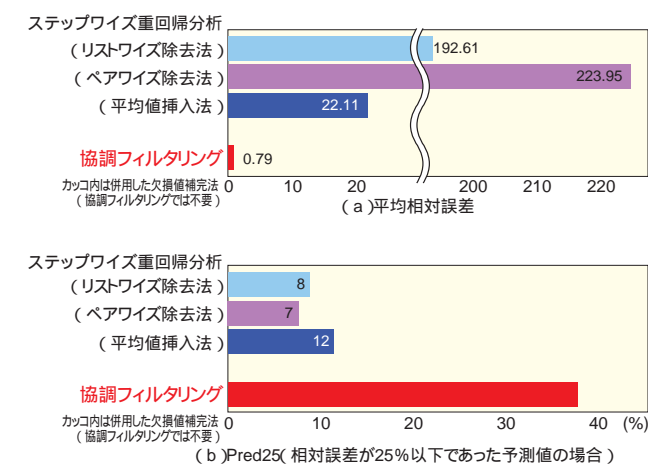


図4 欠損率60%のデータにおける予測誤差

(2) 潜在的意味解析法を用いたソフトウェア分類

1つのモデルであらゆるソフトウェア開発プロジェクトのコスト見積りを行うことに無理があるのと同様に、ソフトウェアやその開発過程に対して唯一無二のベースラインを設定することは現実的ではない。PMBOKでも、ソフトウェア開発プロジェクトは、独自性（個別性）の高いものとされている[12]。ソフトウェアやその開発過程を特性によって分類し、カテゴリー毎にベースラインを設定し、利用する方がより現実的である。

潜在的意味解析法 LSA (Latent Semantic Analysis) とは、自然言語で書かれた文書、単語の類似度を測定する方法である。ベクトル空間モデルに従った手法の1つであるが、ベクトル空間モデルでは検出できない間接的な関連の抽出を可能にしている。ソースコードを文書に、ソースコード上の識別子（変数名、関数名、型名）を単語に、それぞれ対応させることで、ソースコード間の類似度評価、分類（クラスタリング）が可能になる[13]（図5）。

SourceForge.net で公開されている41個のソフトウェアを分類するケーススタディでは、開発環境やプログラミング言語に関する深い知識なしに、複数の観点からの分類が可能であることを確認した。ケーススタディでは、ソースコードを類似度測定の対象としているが、LSAはもともと自然言語で書かれた文書を対象としたものである。要求仕様書や設計ドキュメント、プロジェクト計画書やテスト実施計画書などへの適用も考えられる。ソフトウェア産業全体、組織単位、プロジェクト単位、プロダクト単位など、多様な粒度でのベースライン構築を可能とするものであり「教訓5」に沿った実用性の高い分析方法である。

3.4 活動

(1) ポスドク派遣

EPM適用企業やエンピリカルアプローチを実践する企業などにEASEポスドクを派遣し、データ分析技術の評価を行うと共に、データ分析のニーズを把握し、新たな技術開発の目標設定などを行っている。

例えば、2004年3月から、EASEプロジェクトの中核企業の1つである日立公共システムエンジニアリング株式会社（日立GP）へのポスドク派遣を始めている。日立GPでは、2004年4月から9月まで、ある自社パッケージソフトウェアの開発プロジェクトにおいてEPMによるデータ収集を行った[5]。現在は、3.3項で紹介した「協調フ

ィルタリング」や「潜在的意味解析法」をはじめとして、ポスドクが考案したデータ分析法を収集データに適用し、その結果の評価や解釈を、ポスドクと日立GPのプロジェクト管理者や開発者が共同して実施している。

データ収集はEPMを適用することなどにより、どの企業（組織）においてもある程度共通的に実施できる。しかし、収集データの利用目的やソフトウェア開発プロジェクトが抱える問題は、企業によって異なる。データ分析については、企業ごとに個別に検討し、時間をかけて問題解決や知識獲得を進めていく必要がある。緒に就いたばかりであるが、「教訓9」を踏まえ、ソフトウェア工学の研究と実践の共生を目指した活動を進めている。

(2) ISERNへの参画

ISERN (International Software Engineering Research Network) は、1993年に創設されたエンピリカルソフトウェア工学の産学の研究者集団である[14]。ソフトウェアの開発/利用/管理を支援する技術について、理論面での議論だけでなく、技術の有用性を確かめる実証実験も行っている。現在は、米国、ドイツ、オーストラリア、日本など世界10カ国の37の研究機関が参加し、ソフトウェア工学分野で最も多くの学術論文を発表する組織となっている。2002年からは、著者らの提案により、産学から広く論文を募る国際会議 ISESE (International Symposium on Empirical Software Engineering) を開催し、2003年からはInternational Advanced School of Empirical Software Engineering と呼ばれる研修コースも開設されている。こうした活動は、データ収集技術の向上に資するものであり、教訓3を踏まえたものである。

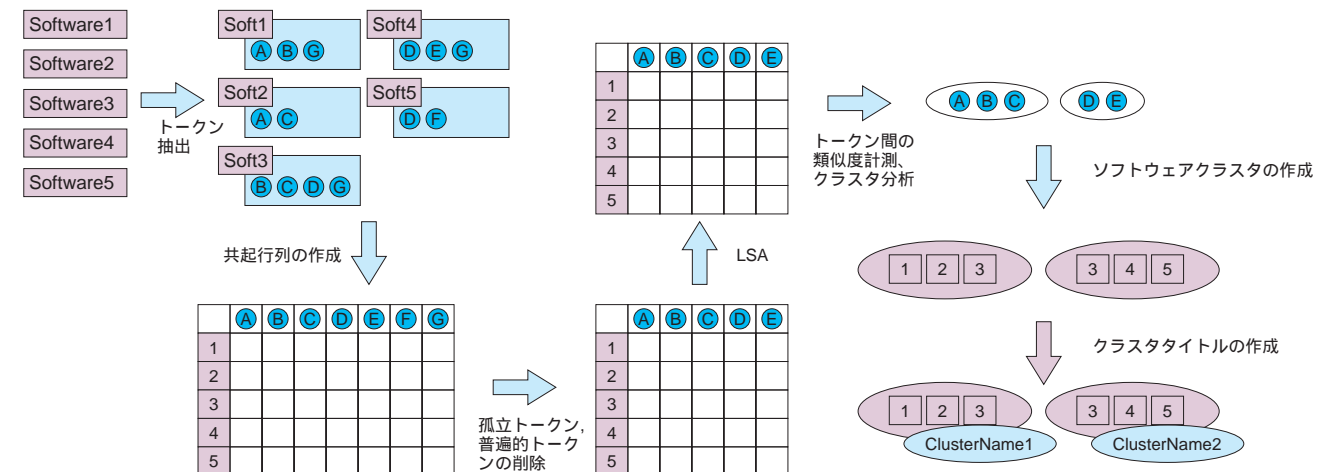


図5 潜在的意味解析法を用いたソフトウェア分類

3.5 組織

(1) エンピリカルソフトウェア工学ラボ

大阪の千里中央に設置している「エンピリカルソフトウェア工学ラボ」は、中核企業4社のソフトウェア技術者とEASEポスドクが同居する場である。まさに教訓10を踏まえた組織である。そうした中からEPMが作り出され、ポスドクと技術者の共著による学術論文も既に発表されている[6]。

EASEプロジェクトでは、工学の原点に立ち返り、地道にソフトウェア開発データを集め、その評価や分析を通じて改善案を提案し開発現場に適用するというサイクルを、できる限り科学的に実践しようとしている。これは、研究者だけではうまくいかないし、開発現場の日々の努力だけでも無理がある。従来の産学連携モデルでは、産と学の間には橋を架けることになるが、EASEでは産と学の自由な交流の場を設ける。中ノ島方式と呼ぶこのモデルで重要なことは、企業に対しては企業の言葉で対応し、学に対しては学言葉で対応する人材の育成である。1人の人間でありながら2つの顔を持ち、必要に応じてそれぞれの言葉を使い分ける。些細なことだが、そうした人材を育てなければ、本当の産学連携などあり得ない。

(2) EPM産業部会

EPM産業部会は、EPMのより広範な普及を促進、サポートを目的として、EASE中核企業や協力企業のソフトウェア技術者を主なメンバーとする組織である。2005年2月から本格的な活動を開始し、現在は9名のメンバーで、EASEプロジェクトおよびEPMへの産業界からの要望を集約し、EPM普及に向けた産学連携の具体的な枠組みやEPMのデータ分析機能仕様などを検討する会合を毎月開

催している。

EPMをはじめとして、EASEプロジェクトが提案するシステムや方法の多くは、産業界のニーズに基づいて開発が始められたものである。しかし、学による検討や具体化の過程を経るにつれ、その狙いや効果がプロジェクト管理者や開発者からはわかりづらいものとなる傾向にある。システムや方法の技術的な説明も重要であるが、教訓13を踏まえるならば、管理者や開発者から見て成功と思われる事例を具体的に示すことも必要となってくる。EPM産業部会では、成功事例の策定についても議論を進めていく予定である。

4 SELからSECへ

3節で示したとおり、EASEはSELの遺した教訓の多くを踏まえたプロジェクトと位置づけることができる。しかし、文部科学省のプロジェクトであることやその予算規模からすると、特に「経営陣の積極的参加に関する4つの教訓」については、十分に対応することが難しい。

これに対して、SECは、より多くの予算を有し、ソフトウェア開発会社が組織するいくつもの団体と太いパイプを持つ[15]。経営陣を巻き込み、「経営陣の積極的参加に関する4つの教訓」を踏まえた、より多様な活動も可能である。SECは、既に、多くの企業の経営陣の支持の下、1,000を超えるソフトウェア開発プロジェクトのデータを集積することに成功している。さらには、ソフトウェアエンジニアリング技術研究組合がSECの協力を得て進めている「先進ソフトウェア開発プロジェクト」における組織横断的なデータ収集にも着手している。ちなみに、SELは、25年間で130余りのプロジェクトのデータしか収集していない。

SELの教訓を単に踏まえるだけでなく、より高次元なエンピリカルアプローチに挑戦することもSECには可能である。例えば、収集データや分析結果の公開や組織間での共有である。

SELはNASAの予算で運営されており、NASAにおけるプロセス改善を支援する組織であった。収集データは原則非公開で、ツール化など成果の普及策もあまり行われなかった。実際、SELの遺した13の教訓には、収集データや分析結果の公開や組織間での共有に直接言及したものはない。これに対して、SECは、文字通り「センター」である。特定の企業のためにデータの収集、分析、蓄積

を行う組織ではない。収集データや分析結果を公開し、測定ツールの開発や提供など普及策も積極的に行い、ソフトウェア開発プロジェクトデータや知識の共有、さらには知識の再生産の中心的役割を担うことも可能である。

収集データや分析結果を公開することに抵抗のある企業は多い。しかし、オープンソースソフトウェアにおいても、ソースコードや開発過程を全てオープンにしたからといって開発に関わる重要な情報がすぐに誰かに筒抜けになるわけではない。情報を獲得し利用するためにはそれなりの知識や技術力が必要であり、明確な目的や強い動機付けがなければ参照すら行われない場合も多い。

それでも抵抗があるというのであれば、収集データや分析結果を（無理に）一箇所に集積するようなことは行わなければよい。コンピュータネットワークで結ばれた環境なら、自分たちが収集したデータや分析結果をローカルに保持することを各企業に許す一方、個人情報の保護や匿名性を確保しつつ、組織横断的にデータを集約し、分析する方式やシステムを実現することは可能である。

例えば、ソフトウェア開発プロジェクトデータの収集や分析に関する「質問回答サイト」を開設してはどうだろうか。質問回答サイトとは、サイトメンバーから寄せられた質問に対して、答えを持つ別のメンバーが回答するコミュニティサイトの一種である[16][17]。インターネット上の膨大な情報の中から必要とする情報をみつけ出すには、検索エンジンとキーワード検索の組み合わせだけでは限界がある[2][18]として検索サイト自体が開設している場合も多い。

質問回答サイトを立ち上げること自体は技術的に難しくないかもしれない。ただし、質問が活発に行われ、多数の有益な回答が寄せられるためには、メンバー間のコミュニケーションを円滑にする次のようなしくみが必要である[18]。

媒体：質問内容をメンバー全員に正しく伝達するためのメディア、データ収集用テンプレートやベースライン、さらには、収集データそのものが共通化されていれば、それらが媒体となるかもしれない。

回答コストの低減：回答コスト（回答に要する工数）が大きいと、回答したくても回答できないということになる。プロジェクト管理者や開発者が、回答のためにわざわざ資料を作成したり整理したりするのではなく、通常の作業やそこで作り出されるプロダクトがそのまま回答や回答の資料になれば、回答コストは大幅

に抑えることができる。

動機付け：有益な回答を確保するためには、ある程度以上の知識や経験を有するメンバーの存在が不可欠である。多くの質問回答サイトでは、一部のメンバーのみが頻繁に回答し疲弊してしまうという事態を避けるために、回答者に報酬を与える仕組みを提供している。

データの分析や知識化までも自動で行ってくれる方式やシステムが理想である。しかし、有能なプロジェクト管理者の知識の多くが暗黙知である現状からすると、様々なソフトウェアプロジェクトにおいて仮説やモデルに基づくデータ分析が可能となるまでにはもう少し時間が必要である。その実現を待って収集データを死蔵することになれば、SECも多くのソフトウェア企業の現状と同じになってしまう。産学官連携の下、様々な可能性への挑戦を期待したい。

5 おわりに

本稿では、米国SELの研究者たちが遺した「プロセス改善におけるエンピリカルアプローチに関する13の教訓」を紹介すると共に、それら教訓と対比させる形でEASEプロジェクトの現状とSECへの期待について述べた。

Basili教授らSELの中心メンバーは、その後、コスト見積りモデルCOCOMOの提案者として著名なB. Boehm教授らとCeBASE（Center for Empirically-Based Software Engineering）を設立し、研究を続けている[20]。自らが得た教訓を踏まえてか、CeBASEでは、SELで収集したデータの一部や新たに開発した測定ツールなどをホームページ上で公開している。名称も「センター」となり、その意味ではSECと同じであるが、NSF（National Science Foundation）がスポンサーとなっていることもあり、ホームページを見る限りでは、大学間連携が中心で、多数の企業が参加するプロジェクトとはなっていない。

エンピリカルソフトウェア工学において、多数の企業との連携が間違ったアプローチであるとは思えない。連携の具体的な形態や進め方についてはまだまだ議論が必要なのかもしれないが、今度ばかりは、CeBASEが教訓を遺してくれるのを待つ時間的余裕はないし、そもそも、多数の企業との連携についての教訓は遺してくれそうにない。ソフトウェアの生産性や品質の向上に実証データ

が必要であることは多くの人が認めるところであり、実際に多数の企業がデータ収集を行ってきている。産学官連携の下、どのような一歩を踏み出すことができるのか、その議論に本稿が役立てば幸いである。

謝辞 本稿の一部は、文部科学省「e-Society基盤ソフトウェアの総合開発」の委託に基づいて行われた研究成果に基づくものである。本稿の完成にあたり、SELの遺した教訓についてご意見をいただいたEASEプロジェクトメンバー、特に楠本真二先生、飯田元先生、門田暁人先生、大平雅雄先生、松村知子氏、森崎修司氏、大杉直樹氏に深く感謝する。また、掲載の機会をくださったSECの皆様にも深くお礼を申し上げる。

参考文献

- [1] “特集：日本のソフト開発力を取り戻せ”，日経コンピュータ，2004年12月13日号，pp.52-69，2004
- [2] A. Ribak, M. Jacovi, V. Soroka: “Ask before you search: Peer support and community building with reach out,” Proc. of ACM conference on Computer supported cooperative work (CSCW'02), New Orleans, Louisiana, USA, pp.126-135, 2002
- [3] “情報サービス産業における受注ソフトウェア開発の技術的課題に関するアンケート調査”，社団法人情報サービス産業協会，2004
- [4] V. R. Basili, F. E. McGarry, R. Pajarski, M. V. Zelkowitz: “Lessons learned from 25 years of process improvement: The rise and fall of the NASA Software Engineering Laboratory,” Proc. of the 24th International Conference on Software Engineering (ICSE2002), Orlando, Florida, USA, May 19-25, 2002
- [5] 井上 克郎，松本 健一，鶴保 征城，鳥居 宏次: “実証的ソフトウェア工学環境への取り組み”，情報処理，Vol.45, No.7, pp.722-728, 2004
- [6] EASEプロジェクトホームページ，<http://www.empirical.jp/>
- [7] 大平 雅雄，横森 励士，阪井 誠，岩村 聡，小野 英治，新海 平，横川 智教: “ソフトウェア開発プロジェクトのリアルタイム管理を目的とした支援システム”，電子情報通信学会論文誌D-I, Vol.J88-D-I, No.2, pp.228-239, 2005
- [8] 村上 弘，飯田 元，松本 健一: “ソフトウェア開発プロセス管理データの収集と活用の支援を目的とした電子ガイドの提案”，電子情報通信学会技術報告，ソフトウェアサイエンス研究会，No.SS2004-41, pp.43-48, 2004
- [9] JIS X0141:2004，ソフトウェア測定プロセス。（ISO/IEC 15939:2002, Software engineering_Software measurement process.）
- [10] “CMMIモデル・公式日本語翻訳版”，Software Engineering Institute, <http://www.sei.cmu.edu/cmmi/translations/japanese/models/index.html>
- [11] J. Kromrey, C. Hines, “Non-randomly missing data in multiple regression: An empirical comparison of common missing-data treatments,” Educational and Psychological Measurement, Vol.54, No.3, pp.573-593, 1994
- [12] 角田 雅照，大杉 直樹，門田 暁人，松本 健一，佐藤 慎一: “協調フィルタリングを用いたソフトウェア開発工数予測方法”，情報処理学会論文誌（産学連携論文特集），（2005年5月掲載予定）
- [13] “プロジェクトマネジメント知識体系ガイド（PMBOKガイド）2000年版”，プロジェクトマネジメント協会，2000
- [14] 川口 真司，松下 誠，井上 克郎: “潜在的意味解析法LSAを利用したソフトウェア分類システムの試作”，情報処理学会研究報告，Vol.2003, No.22, pp.55-62, 2003
- [15] ISERNホームページ，<http://www.iese.fhg.de/ISERN/>
- [16] SECホームページ，<http://www.ipa.go.jp/software/sec/index.php>
- [17] Experts Exchange ホームページ，<http://www.experts-exchange.com/>
- [18] goo教えて！gooホームページ，<http://oshiete.goo.ne.jp/>
- [19] J. Teevan, C. Alvarado, M.S. Ackerman, D.R. Karger: “The perfect search engine is not enough: a study of orienteering behavior in directed search”, Proc. of 2004 Conference on Human Factors in Computing Systems (CHI'04), Vienna, Austria, pp.415-422, 2004
- [20] CeBaseホームページ，<http://www.cebase.org/>

ソフトウェア開発の 生産管理に基づく見積りモデル

The Estimation Model Based On Development
Management For Software Development



株式会社ジャステック
常務取締役 営業本部本部長
太田 忠雄

JASTEC Director, EVP and CMO
Tadao Ohta

株式会社ジャステックは、創業以来、役務提供を前提とした人月契約から脱却し、一括請負契約を拡大すること、および生産性原理に立脚した能力主義を実践するために、独自の生産管理システムを構築した。

生産管理システムの中心となるのがソフトウェア開発の見積りモデルである。見積りモデルは、開発計画の精度を高め、開発計画達成へのコントロールを可能にし、さらには開発プロセス改善の機会を創出するために必要である。

そのような要求を満たす見積りモデルとして、ジャステックは生産物量と生産性を変数とした基本アルゴリズムを考案した。生産物量とは、ソフトウェア開発の各工程で作成する生産物の量であり、生産性とは生産物単位のスピード、あるいはコストである。基本アルゴリズムは、生産物量見積り方式および生産性見積り方式から成り立っている。また、それぞれの方式において、開発環境の違いや品質要求の多寡による変動を吸収する「環境変数」と呼ぶパラメータを導入している。さらに、基本アルゴリズムを拡張し、改造型開発の見積りモデルおよび仕様変更の見積りモデルを構築した。

本稿がソフトウェア開発産業発展の一助となれば幸いである。

We have produced our original Management System for Software Development, based on the productivity theory for software development. A fundamental part of this system is the estimation model of software development. It allows us to raise the accuracy of development planning, to make it controllable and to get opportunities for process improvement.

The model has the basic algorithm, which consists of the product volume estimation method and productivity estimation method. Each method introduces change factors, called "environment parameters" in order to correct the differences of development environments and also changes of demanded quality. Moreover we have expanded the basic model to apply the estimation models to modification and specification changes.

We feel happy if this paper would be any help for the growth of the software development industry.

Key Words & Phrases : 見積りモデル, 生産物量, 生産性, 環境変数, 仕様変更, CMMI
Estimation model, Volume, Productivity, Change factors, Specification change, CMMI

1 はじめに

ジャステックは独立系のソフトウェア開発会社として、ソフトウェア開発および販売を専業としている。特に、開発関連業務の多角化ではなく、開発分野の総合化と流通化を図り、システムインテグレーション市場の拡大を通して情報社会に貢献することを目的に取り組んでいる。

わが国での「勝ち組」の製造業企業は、パートナー会社との生産性原理に基づく相互協力を前提に、お互いの創意工夫を促し、コスト削減と品質改善目標を達成してきた。ジャステックは製造業をお手本に、ソフトウェア開発を製造と捉え、ソフトウェア開発の生産管理を定量的に行うために、独自の生産管理システムを導入してきた。受託ソフトウェア開発の価格は量と正の相関関係にあ

ることを前提にしている。本論の見積りモデルは生産物量と生産性を変数とした基本アルゴリズムを適用している。一般に、ソフトウェアは可視化しにくいといわれるが、決して可視化できないわけではない。ジャステックではソフトウェア開発工程毎の生産物を開発量として可視化している。生産物量とは、ソフトウェア開発の各工程で作成する生産物、つまり設計書、プログラムソース、テストデータ等の量である。また、生産性とは生産物量単位、つまり文字数 (kC)、ソースコード数 (kLOC)、テスト項目数等を作成するために要する時間数、あるいはコストである。生産物を開発量として捉えるには、アクティビティ毎の記述項目、記述水準およびカウント方法などの標準化を図る必要がある。アクティビティとは、工程毎の生産物をさらに分別し、作業の割り振りを容易にするための作業単位である。記述水準とは、分別した

生産物の記述の深さを規定する。基本アルゴリズムは、生産物量見積り方式および生産性見積り方式から成り立っている。また、それぞれの方式において、開発環境の違いや品質要求の多寡による変動を吸収する「環境変数」と呼ぶパラメータを導入している。

本稿では、見積りモデルの基本アルゴリズムと生産管理システムについて述べる。第2節では生産性原理論から始まった生産管理システムの導入背景を説明し、第3節で顧客と開発ベンダにとっての見積りの重要性に関して提言している。そして、第4節では基本アルゴリズム、拡張モデル（改造型開発と仕様変更の見積りモデル）および生産管理システムを中心に解説し、さらに、第5節で見積りモデル、生産管理システムを維持運用する体制・役割分担・企業文化を紹介する。最後の第6節ではモデル運用上での課題と今後の取り組みについて述べる。

2 取り組みの背景

2.1 経緯

市場原理と独創性を根底とするグローバルな企業競争が激しさを増す中、わが国のソフトウェア開発業界では、見積り技術を含む生産管理技術および品質管理技術を基盤にした、一括請負契約を標榜するシステムインテグレーション企業が十分には育っていない。

労働集約型の派遣契約では、ソフトウェア開発の見積り技術が向上せず、生産性を上げる努力を社員も経営者もしなくなる。といった声から取り組みが始まった。

ジャステックは創業以来、労働集約型ではなく知識集約型のソフトウェア開発専門企業を標榜してきた。

昭和58年に労働者派遣法成立の機運が高まる一方で、脱マンパワーリースを率先垂範すべく、創業以来培ってきたプロジェクトマネジメント技術を集大成し、ソフトウェア開発の生産管理システムを構築した。生産管理システムの導入は生産性原理の具現化を促した。そして、見積りの基礎となる開発計画および出来高管理の定量化は、定量的プロジェクト管理の第一歩であった。さらに、平成8年には独自の生産管理システムを基軸に、品質保証の国際標準であるISO 9001を取り入れ、組織レベルでの規定を整備し、運用を徹底した。平成14年1月にSEI（米国カーネギーメロン大学のソフトウェア工学研究所）から、ソフトウェア・プロセス改善の国際標準的な指標であるCMMI（能力成熟度モデル統合）「V1.1」が、公開評

価版として発表された。CMMIのレベル5は、ジャステックでの生産管理システムのコンセプトと符合することを知り、平成15年には、CMMIのレベル5を融合した独自の品質保証体系を確立し、運用の徹底を図り、今日に至っている。

2.2 トピックス

ソフトウェア開発の取り組みに関わる最近のトピックスを2点、以下に挙げる。

平成15年に東京証券取引所の市場第1部へ上場すると共に、全社（総務経理部門および営業部門を除く）を対象としたソフトウェア開発プロセスの能力成熟度を評価する国際標準的な指標であるCMMIの最高成熟度レベル（レベル5）を達成している。

アセスメント内容は、参照モデルが「CMMI-SE/SW/IPPD/SS Version 1.1 連続表現」である。アセスメント手法が「SCAMPI Version 1.1」であり、アセスメント範囲は「23個のプロセスエリア（2個のハードウェアに関するプロセスエリアを除く）」を対象としている。

独自でのソフトウェア開発の生産管理および品質改善の取り組みは、結果として業界屈指の高収益の達成を可能とした（平成16年度の売上高経常利益率16.8%）。また、これらの取り組み、および健全経営の評価を得て、平成16年度情報化促進貢献企業として経済産業大臣賞を受賞した。

3 見積りの重要性

ソフトウェア開発を取り巻く環境は、経営とITの呼应性への高まり、システム構造の複雑化およびシステム化技術の多様化等が進展してきている。このような環境下、顧客と開発ベンダの経営において、見積りは益々、重要性が高まっている。一般に、顧客経営での重要性は、妥当な投資額の把握および見積りミス（品質、コスト、納期）によるビジネス機会喪失の回避がある。開発ベンダでの重要性は、健全な企業収益の確保および見積り説明責任による顧客からの信頼の確保がある。

次に、ジャステックでの2つの見積りの重要性について述べる。

1つは生産管理システムの導入効果を高めるための見積りの重要性である。生産管理システムの導入効果を高めるには見積り精度を向上させる必要がある。例えば、

見積りはプロジェクトのコントロール範囲内に精度を精錬させておくことである。ジャステックの見積りは開発計画の量、生産性、品質、納期などと整合する。見積りはプロジェクトチームおよび個人の目標を明示し、生産物量による出来高などの監視指標に基づき、プロジェクトをコントロールし、目標の達成を可能にしている。さらに、生産管理システムは、生産性や品質などの見積りのベースラインを設定するのに、プロセス資産の蓄積・分析を通して、その一翼を担っている。

2つ目は生産性向上に関する見積りの重要性である。生産性向上とは、顧客との相互協力による生産性向上および技術者個人の生産性向上を示す。見積りは顧客との共通語である。例えば、見積り段階から量および生産性の変動パラメータ（外部環境変数）を開示して、プロセスの改善点を客観的に共有する。プロセス改善効果は、生産性向上度合いを定量化することで、相互に享受可能とした。次に、技術者個人の生産性向上である。ジャステックは能力主義を実践している。そのために、評価尺度の1つに個人の生産性と品質がある。見積りモデルは、プロジェクト固有の変動パラメータを可視化し、客観的な個人の生産性と品質を捉えることを可能としている。

4 独自の生産管理に基づく見積りモデル

4.1 見積りモデルの基本アルゴリズム

(1) コスト見積りモデル

ジャステックが考案した生産物量と生産性を変数とする新規開発の基本アルゴリズムを示す。本アルゴリズムはソフトウェア製品を新規に開発する場合に適用するものである。ある工程 i の生産物量を V_i 、生産性を P_i 、標準生産物量を V^{B_i} 、生産性のベースラインを P^{B_i} で表現すると、コスト C_i は次式で求まる。

$$C_i = V_i \times P_i = V^{B_i}(1 + a_i) \times P^{B_i}(1 + b_i)$$

$$\text{ただし } a_i : (\quad i_j) \quad b_i : (\quad i_j)$$

a_i 、 b_i は、それぞれ V^{B_i} および P^{B_i} に対して開発環境の違いや品質要求の多寡による変動を吸収する「環境変数」と呼ぶパラメータである (a_i : 生産物量環境変数, b_i : 生産性環境変数)。図1に示す通り、 a_i 、 b_i は品質特性と環境特性から影響される独立した変動要素 (i_j, i_k) から構成されている。品質特性は、 V^{B_i} および P^{B_i} の双方に影響し、環境特性は P^{B_i} のみに影響する。本モデルは、ソフトウェア開発で作成する生産物毎に適用し、プロジェクトのコストは複数の生産物の総和となる。なお、 V_i を

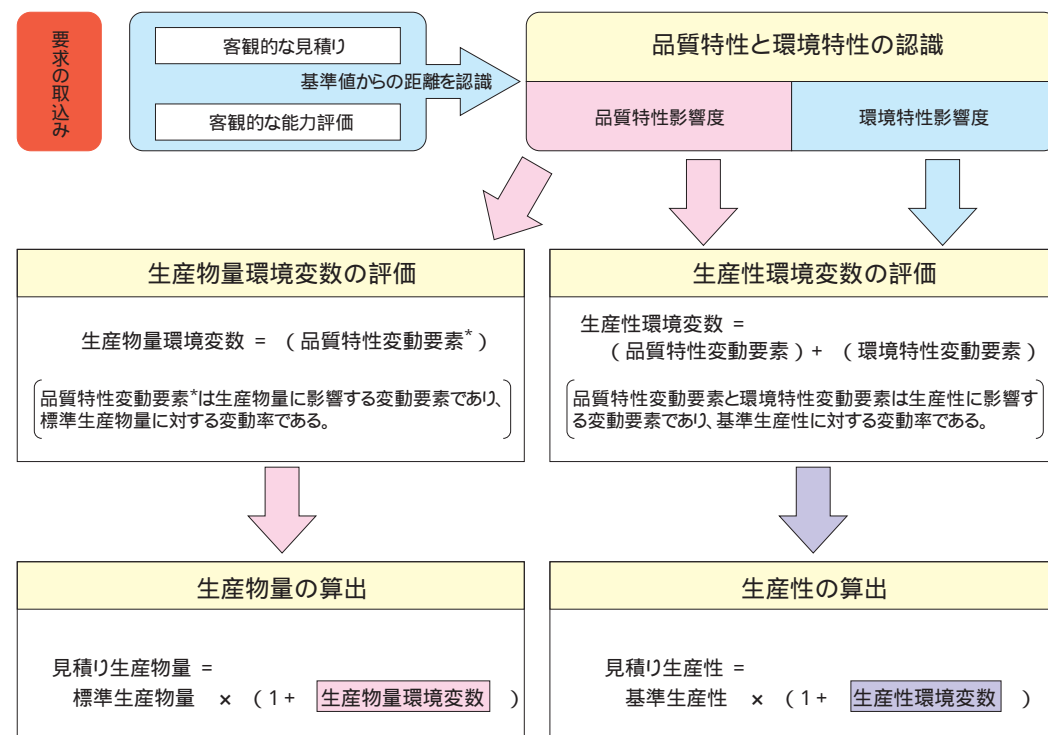


図1 生産物量と生産性の算出式

求めるアルゴリズムを生産物量見積り方式、 P_i を求めるアルゴリズムを生産性見積り方式と呼ぶ。

(2) 生産物量見積り方式

ある工程 i の生産物量 V_i の求め方を、図2に示す。良く規格化された（記述項目、水準、書式、表現方法などが定められていること）基本設計書の量は、ソフトウェア要求の量を測る尺度としてよく用いられる「ファンクションポイント (FP)」との相関が高いことが経験的にわかっている。また、各工程の生産物量の間にも高い相関関係があることに着目し、上位工程で作成する生産物量から下位工程の生産物の量を算出するアルゴリズムを考案した。

$$V^{B_i} = V^{B_{i-1}} \times H_i$$

ただし H_i : 生産物量変換係数

生産物量の変動パラメータである生産物量の環境変数を a_i として表現すると、生産物量は次式で求まる。

$$V_i = V^{B_i} \times (1 + a_i)$$

ただし $a_i : (\quad i_j)$

任意の生産物は複数の記述項目から構成され、その中には要求の量または実装結果であるソースコードなどの量との相関が高いものや低いものが混在している。本方式では、前者を標準生産物量 (V^{B_i}) と呼ぶ。後者は、表1に示す品質特性変動要素評価表に基づき生産物量の環境変数 a_i を算出して、 V^{B_i} を調整する。品質特性は日本工業規格 (JIS X0129) に基づき品質特性および副品質特性

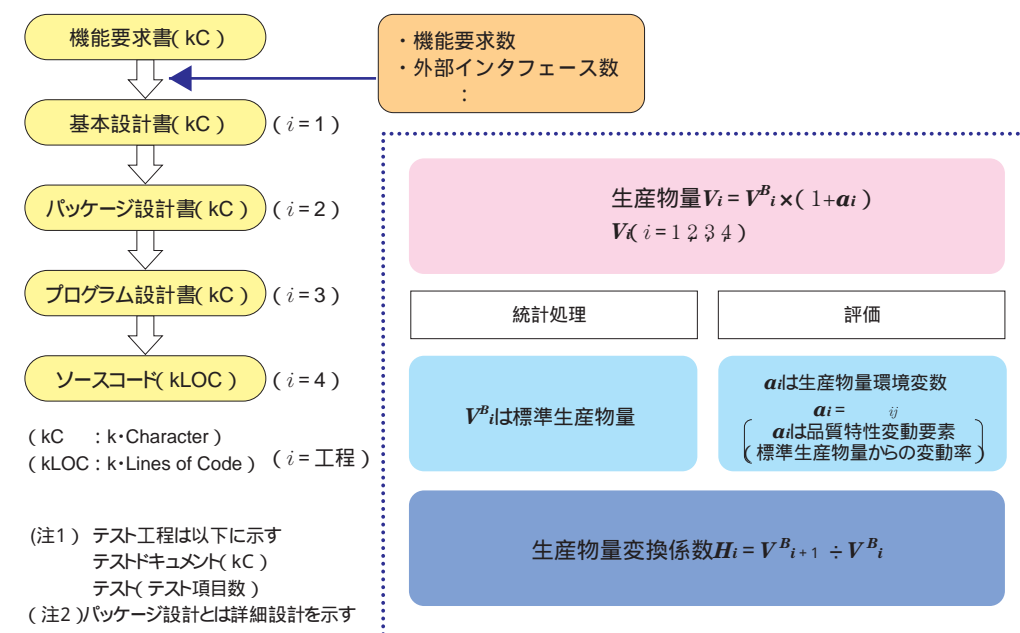


図2 生産物量見積り方式のアルゴリズム

を識別し、尺度および数量化方法を定めており、顧客の要求水準を確認して決定する。

生産物量に影響する品質特性は4個あり、さらに品質特性を細分化した副品質特性が9個ある。

(3) 生産性見積り方式

モデルを単純にするために個人の能力差による生産性の相違を吸収して、生産性を生産物の単位量当たりのコストとした。生産性の変動パラメータである生産性の環境変数を b_i として表現すると、生産性は次式で求まる。

$$P_i = P^{B_i} \times (1 + b_i)$$

ただし $b_i : (\quad i_j)$

生産性のベースライン P^{B_i} は、生産性の環境変数 b_i による変動を除去した後の実績生産性に基づく平均値を基準にしている。次に、開発ベンダの立場から、環境変数の外部環境変数および内部環境変数について述べる。

外部環境変数には「顧客の特性による変動」および「プロジェクトの特性による変動」があり、ソフトウェアの価格に反映する。

内部環境変数は「開発ベンダの能力による変動」であり、ソフトウェアの価格に反映しないが、プロジェクトの実行計画には反映しなければならない。

生産性の環境変数の具体例を、表1 (品質特性変動要素評価表) および表2 (環境特性変動要素評価表) に示す。生産性に影響する環境特性は6個あり、「コミュニケーション特性」、「入力情報特性」、「開発環境特性」、「業務特性」、「ハードウェア特性」および「ソフトウェア特性」である。さらに、環境特性を細分化した副環境特性は13個あり、それぞれに外部環境変数および内部環境変数に区分している。「コミュニケーション特性」を例にすると、その配下には「組織複合度」、「顧客窓口特性」、「システム規模」および「コミュニケーション基盤」の4個の副環境特性を定めている。

次に、生産性に影響する

表1 品質特性変動要素評価表の例

品質特性	副品質特性	影響水準			評価 (変動率)
		変動要素	レベル	具体値	
機能性	合目的性	要求仕様の網羅性 (実記述文字数÷標準記述文字数)	1		基準生産性 「ベースライン」 に対する 増/減(%)
			2	95%~100%	
			3	75%~95%未満	
			4	50%~75%未満	
			5	50%未満	
	正確性	レビュー充実度 (標準レビュー 工数比率との 乖離度合)	1		基準生産性 「ベースライン」 に対する 増/減(%)
			2	1倍	
			3	1.5倍	
			4	2.0倍	
			5	3.0倍以上	
	テスト密度 (標準テスト項目 数との乖離度合)	1		標準生産物量 に対する 増/減(%)	
		2	1倍		
		3	2.0倍		
		4	4.0倍		
		5	8.0倍以上		
付加作業数 (連続立会、移行 作業等)	1		基準生産性 「ベースライン」 に対する 増/減(%)		
	2	付加作業数0			
	3	付加作業数1			
	4	付加作業数2			
	5	付加作業数3以上			

日本工業規格(JIS X0129)

表2 環境特性変動要素評価表の例

環境特性	副環境特性	影響水準			評価 (変動率)
		変動要素	レベル	具体値	
コミュニケーション特性	組織複合度	窓口数 「意思決定者数」 (当社がコーディネータとして担当する相手の窓口数)	1		基準生産性 「ベースライン」 に対する 増/減(%)
			2	単一	
			3	2~3個	
			4	4個以上	
			5		
	顧客窓口特性	正確な回答の 期限厳守度	1		基準生産性 「ベースライン」 に対する 増/減(%)
			2	98%~100%	
			3	90%~98%未満	
			4	80%~90%未満	
			5		
	システム規模	システム開発規模の 大きさ (実際の適用は開発 期間との相関が考慮 される)	1	10ks以内	基準生産性 「ベースライン」 に対する 増/減(%)
			2	10ks~50ks以内	
			3	50ks~300ks以内	
			4	300ks~800ks以内	
			5	800ks以上	
コミュニケーション基盤	物理的な要因 (拠点分散等)	1		基準生産性 「ベースライン」 に対する 増/減(%)	
		2	1ヶ所		
		3	数ヶ所		
		4	数ヶ所(海外含む)		
		5			

品質特性は5個あり、さらに品質特性を細分化した副品質特性が14個ある。

生産性の環境変数は、品質特性と同様に尺度および数量化方法を定めており、顧客の特性水準およびプロジェクトの特性水準を確認して設定する。この過程で、見積り対象のプロジェクトが抱えるリスクが明らかになり、顧客とリスクを共有し、相互協力による生産性改善策を講じることで効果を挙げている。

生産物は、複数の記述項目から構成されているので、生産性も記述項目に対応させ保持しており、生産性の環境変数は各記述項目の生産性に対して作用させている。

なお、各工程での外部環境変数とベースラインからの変動幅が、±30%以内になることを経験値から得ている。ただし、内部環境変数の混入除去、組織標準の精練および運用の徹底を図った上での値である。

4.2 改造型開発の見積りモデル

本モデルは新規開発の見積りモデルに比べ、成熟度の点から検討すべき課題を含む。ジャステックは、今後とも研究と実証を通して本モデルの精練を図っていく。ここでは、現時点での改造型開発の見積りモデルについて述べる。

改造型開発と新規開発との主要な相異点は3つある。1

つは、改造型開発では要求を実現する手段が改造対象システムの内部構造に制約されるため、図3に示すように改造要求による影響範囲を識別し「改造正味規模」、「母体規模」および「巻き込み規模」を特定しなければならない。2つ目は、新規開発では次工程への指示書と製品の設計書(製品仕様書)とが同一であるが、改造型開発では改造した結果の製品仕様書とは別に、改造事項を特定する指示書(改造設計書)を作成する。改造設計書の生産物量は、「改造正味規模」、「母体規模」および「巻き込み規模」をパラメータとして見積ることとした。設計作業の負荷と相関が高いのは改造設計書なので、改造設計書を設計の生産物としてその量を見積る。なお、製品仕様書は別の生産物として扱う。3つ目は、改造対象システムの調査が必要であり、場合によっては改造対象システムの設計書をソースコードなどから復元する必要があるということである。「改造対象システムの調査に対応する生産物量には、調査対象の生産物量を割当て、改造対象システムの設計書の復元作業に対応しては、復元仕様書を生産物として追加した。」

改造型開発における見積りモデルの環境変数は、新規開発の見積りモデルの環境変数を併用すると共に、改造母体に対する練度など、改造型特有の環境変数を追加している。

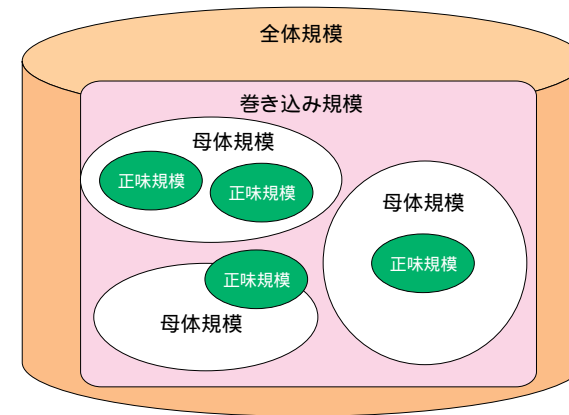
4.3 仕様変更見積りモデル

ジャステックでは契約にあたり、発注者(顧客)の決断を必要とする項目、および発注者との相互合意を必要とする項目について提示している。特に、仕様変更見積りに関する発注者の決断項目には、仕様変更に伴う量(追加量、棄却対象量等)と変更タイミングがある。相互合意の項目には、仕様変更の認定基準と仕様変更に伴う量(正味棄却量)がある。

次に、「仕様変更量」とは何か、について述べる。図4に「仕様変更量」の構造と算出式を示す。

ここで、「当初量」とは仕様変更がまったくなかった場合の見積り量である。「変更棄却対象量」とは、仕様変更によって開発不要となった部分の見積り量で、「変更正味棄却量」とは、仕様変更により途中まで開発したが開発不要となり実際に捨てるであろう量である。また、「変更追加量」とは仕様変更により追加となるであろう量である。開発完了時に最終的に納品される量が「最終実現量」であり、開発コストに比例するのが「開発量」である。

注目すべきは「変更正味棄却量」である。この量は、仕様変更の発生するタイミングによって、同じ変更でも大きく変動する。つまり、基本設計時点で発生すれば、変更により不要となり棄却するのは基本設計書の該当部分だけであるが、システムテスト時点で発生すれば、基



改造正味規模
改造要求に対して、実際に修正および追加するソース規模
母体規模
改造正味対象となるソフトウェア・アイテムの改造前ソース規模
(改造正味部分を確定する為に調査・分析の対象となる規模)
巻き込み規模
改造正味に関係するシステム全体でのダイナミックな実行ルートに関わるソフトウェア・アイテムのソース規模
(テストの対象範囲を確定する)
(注意)ソフトウェア・アイテムはプログラム(コンパイル単位)を指す

図3 改造型開発の規模構造

本設計書からシステムテストまでの全ての工程の生産物を棄却することとなる。

4.4 見積りモデルの選択

(1) 開発形態による見積りモデルの適用方法

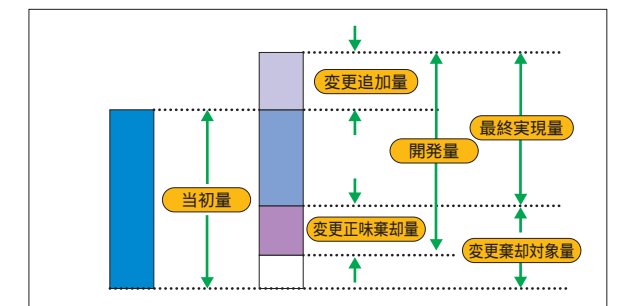
ジャステックの見積りモデルはウォーターフォール・ライフサイクル・モデルの適用を前提としている。スパイラル・モデルには、開発サイクル毎に本モデルを適用し、インクリメンタル・モデルにはインクリメンタルの段階毎に本モデルを適用する。また、既存システムの保守作業は、個々の保守案件毎に本モデルを適用している。規模の小さい案件では、各工程で作成する生産物量をソースコードの量に代替することで、簡易化を図っている。

(2) 見積り時点による適用方法

本モデルでは、プロジェクトの途中で順次完成する生産物量に基づき、簡単に生産物量の再見積りができる。また、ソースコード量と基本設計書の量との関係式も導出でき、ソースコード量から基本設計書の量を推定する場合に利用できる。基本設計が完了していない場合には、次の方法のいずれかを選択して、まず基本設計書の生産物量を見積る。

類似のシステムや再構築対象の既存のシステムが存在する場合など、確からしいソースコード量を計測できる場合は、ソースコード量から基本設計書の量を見積る。

ユーザの要求を確認して、システムへの入力、出力、機能を洗い出し、これを根拠に基本設計書の量を求める。この場合の精度は、洗い出しに投下する作業に依存する。通常は「および」を併用して、見積りの精度を高め



$$\text{開発量} = \text{当初量} - \text{変更棄却対象量} + \text{変更追加量} + \text{変更正味棄却量}$$

$$\begin{aligned} \text{最終実現規模} \\ \text{変更棄却対象量} &= \text{当初量} \times \text{変更棄却対象率} \\ \text{変更追加量} &= \text{当初量} \times \text{変更追加率} \\ \text{変更正味棄却量} &= \text{当初量} \times \text{変更棄却対象率} \times \text{完成率} \end{aligned}$$

図4 仕様変更量の構造と算出式

ている。

ただし、基本設計が完了するまでは、見積精度の低下を防ぐことはできないので、「企画・調査段階」、「要件定義中」、「要件定義完了時点」、「基本設計完了時点」と数段階の見積りを定めている。実務上では顧客との合意に基づき段階を決定し、契約を取り交わしている。

4.5 プロジェクト実績の蓄積と活用

(1) プロジェクトマネジメントでの活用

ソフトウェア開発を成功させるためには、開発途中の生産物量および生産性を日々監視し、それが見積りと乖離しないようコントロールする必要がある。ジャステックには個々の技術者が日々の生産物量と、それに要した工数を入力し、リアルタイムに出来高を把握する仕組みがある。ジャステックでは生産実績の分析手法を定め、具体的なプロセス改善を図っている。事例として、開発量の予実差異分析は、生産物量環境変数と仕様変更量の変動量に分解し、生産性の予実差異分析は生産性環境変数の変動値に分解する。この分析によって、予実差異の根本原因を可視化している。さらに、仕様変更量、環境変数の中で、変動が懸念される項目を開発リスク項目と

して管理し、その顕在化の監視手順、防止方法および軽減化手順をリスク対策事項として定め、開発計画に組み込んでいる。

(2) 技術者個人の生産性と品質の把握

技術者個人の生産性と品質の把握は、能力主義を实践する上で必要であることを既に述べた。個人の生産性と品質を把握するには、生産実績の入力機能が必要である。さらに、ジャステックの生産管理システムは、プロジェクト固有の環境変数を可視化すると共に、被指示者、指示者といった開発体制面の相互作用を吸収するように配慮している。

5 体制・役割分担・企業文化

ジャステックは創業以来ソフトウェア開発部門を製造部（製造本部）という組織名にしている。本論文では、既に製造業をお手本にしていることを述べた。

独自の生産管理システムの実施体制については、図5に示す。

本節では見積りモデルに関係する各部門の体制と役割を中心に述べる。

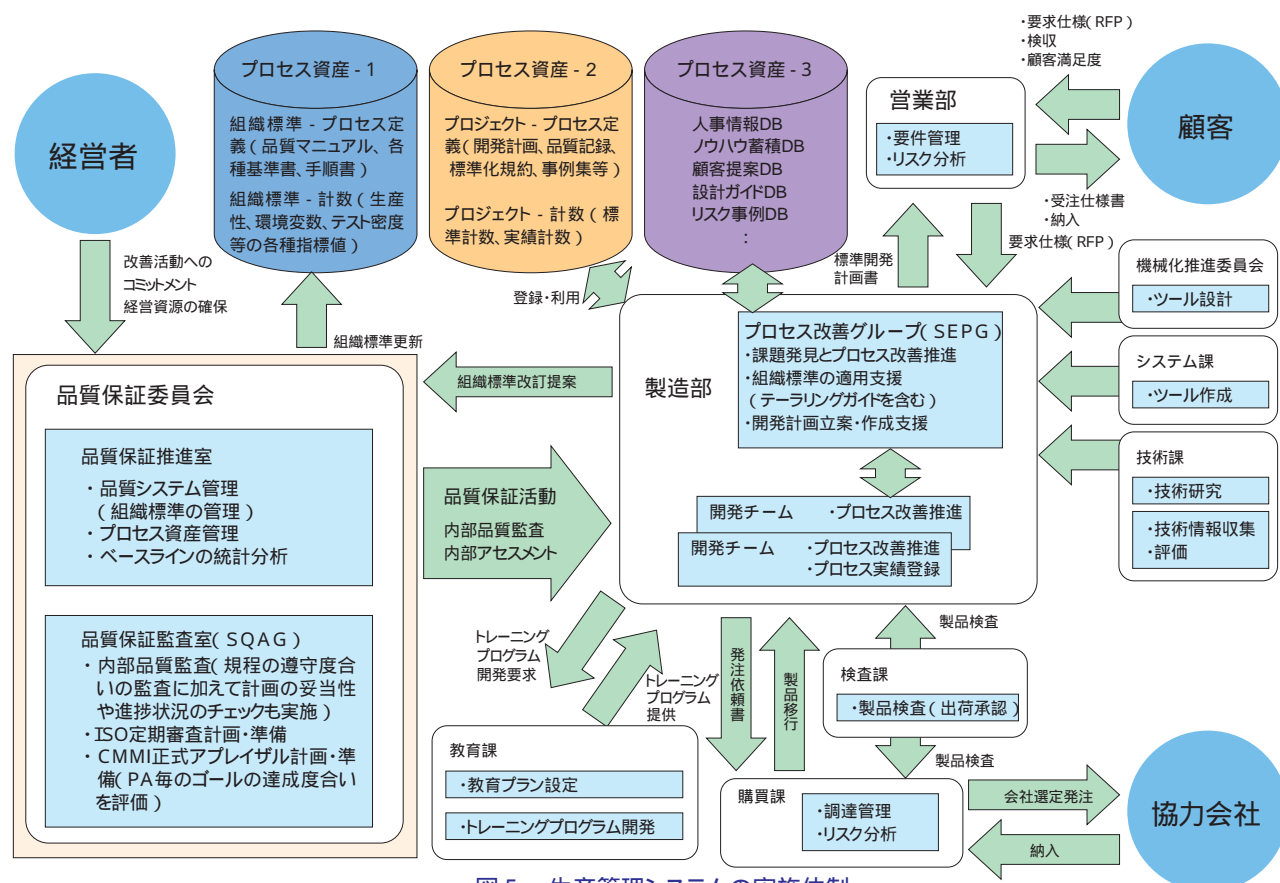


図5 生産管理システムの実施体制

ジャステックの見積りモデルには、顧客向けの外部見積り（標準見積り）と社内向けの内部見積り（標準開発計画）がある。見積りの中心となるのが標準開発計画である。

標準開発計画は顧客のRFP（提案依頼書）に基づき、営業部門と製造部門とが共同でサーベイ作業を行い、作成している。標準開発計画は環境変数を配慮した量、生産性などを基準に見積っている。標準開発計画は製造部門の達成責任になる。

標準見積りは顧客との共通語とするために、標準開発計画の量、生産性を継承することが前提にある。

それゆえ、営業部門の責任は価格折衝責任を除くと、継承責任が重要になる。継承責任とは量、生産性と共に、変動パラメータの環境変数を顧客に公開し、相互共有と協力関係を構築することである。

標準開発計画とその実績は、プロジェクトのプロセス資産としてデータベースに蓄積している。製造部門ではプロジェクトチームの他に、SEPG（ソフトウェアプロセス改善グループ）を設置し、製造部門に共通するプロセスの改善およびプロセス資産の活用促進等を行っている。

次に、製造部門を牽制し支援する品質保証関連の組織について述べる。1つは検査課で、成果物の検査を行う専門組織である。検査課は成果物の検査以外に、プロジェクトの計画と実績との乖離を統計分析し、製造部門では是正処置の動機付けを促している。2つ目の組織は品質保証委員会である。品質保証委員会は、品質マニュアルなどの組織標準およびプロセス資産を管理推進すること、ならびにプロセスの監査を行う専門組織である。特に、見積りの基礎となる生産性のベースラインは本組織で統計分析し、製造部門にフィードバックしている。

最後に、各部門の見積りに関連するプロセス改善目標は、経営者のマネジメントレビューに基づき予算化し、達成度合いを監視している。

6 現在の課題と今後の取り組み

(1) 企画・要件定義工程での見積り精度の向上

ジャステックの見積りモデルも、他のモデル（ファンクションポイント法など）でも、企画・要件定義工程と以降の開発工程の見積根拠を可視化し、精度を向上するという課題がある。ジャステックの課題への取り組みは、本稿の生産管理に基づく見積りモデルを基軸に行う。現

状の見積りモデルは、企画・要件定義工程でのアクティビティ、生産物（記述項目、水準、書式、表現方法）、環境変数等が十分とはいえない。今後、顧客の協力を得て、継続的に取り組んでいきたいと考える。

(2) 簡易版見積りモデルの開発および普及

本稿での見積りモデルは、詳細に積算するため、一般企業が初期導入する場合に障壁が高いと認識している。また、顧客視点からのご意見（外部環境変数項目の追加等）もある。その様なことを配慮して、顧客企業や開発ベンダなどで広く利用可能な、簡易版の見積りモデルの開発を検討している。

7 おわりに

本稿で述べてきたモデルや方式は、ソフトウェア開発の見積りおよび生産管理を实践するためのツールである。ソフトウェア開発は、「優れたツールを導入しても即効的に効果が得られない」といった特質がある。つまり、ツールを利用する基盤が整って初めて効果を発揮する。基盤とは、ソフトウェア開発の特質を共有した顧客との信頼関係、経営者のリーダーシップおよび透明性の高い企業文化などである。

今後のソフトウェア開発を取巻く環境は、益々、複雑化し不確実性を増加させ、リスクが拡大していくと思われる。対応策として、見積りモデルのベンチマーキングを提案したい。ベンチマーキングとは見積りモデルを、実際のプロジェクトで試行し、その実績および適用上の教訓を公開することで、広くノウハウを共有することである。

謝辞 本稿の執筆に快く承諾し支援をいただいた、ジャステックの神山茂社長、庭野幸夫等の諸氏に感謝する。また、投稿の機会を提供いただいた経済産業省ご関係者と、独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センターの見積手法部会をはじめ、関係者の方々に深くお礼申し上げます。

参考文献
 [1] Capability maturity Model Integration (CMMISM), CMU/SEI-2002-TR-001 ~ 004
 [2] IDEALSM A User's Guide for Software Process Improvement, CMU/SEI-96-HB-001
 [3] Production Management Seminar, ソフトウェア開発の生産管理における共通基盤の確立(JASTECC)

組み込みシステムとユーザビリティ工学

Incorporating usability engineering in development of embedded systems

小樽商科大学商学部社会情報学科
助教授(工学修士)
NPO法人人間中心設計推進機構
平沢 尚毅



Otaru University of Commerce Dept. of Information and Management of Sciences
Associate Professor (Master of Engineering)
Human Centered Design Organization, NPO
Naotake Hirasawa

ユーザビリティ工学が目指す品質概念として、利用品質という考え方を説明した。利用品質は、組み込みシステム市場の今後の発展を考える上で、欠かせないものである。その上で、高い利用品質の水準を達成するために、組み込みシステム開発が直面する課題に対して、ユーザビリティ工学が貢献する可能性が大きいことを示した。組み込みシステム開発プロセスに、ユーザビリティ工学を取込むことによって、国際的に競合できる新しいものづくりを確立することができる。そのためには、人材育成に取組み、そのための組織的な基盤作りを緊急にしなければならない。

Concept of Quality in use was introduced as quality concept aimed by usability engineering. It could be essential for embedded system manufactures for expanding the systems. Accordingly, to keep and improvement of the high standard, quality usability engineering will take an important part of solving challenges of embedded system development. Incorporating usability engineering into embedded system developing processes will make international competitive development capability raised. Human resource development with is first key step for promoting it and organizational infrastructures must be established.

Key Words & Phrases : ユーザビリティ 人間中心設計 要求仕様定義 UI設計 ユーザビリティ評価
usability human-Centred Design Requirement Definition UI Design usability Evaluation

1 はじめに

この数年、書籍においてもユーザビリティに関するものが増え、ユーザビリティという言葉が浸透するようになってきている。特に、Webユーザビリティに関するガイドラインが増えている。ユーザビリティに関連したものと、ユニバーサルデザインやアクセシビリティを含めると、急激に社会的に浸透しつつあることがうかがえる。2003年には、情報処理学会誌において、ユーザビリティに関する特集記事が組まれている[1]。

では、ユーザビリティとは何であるのか。実は、明確なコンセンサスがあるわけではない。

一般的には、ニールセンによる定義[2]が引用されることが多い。彼は、ユーザによるシステムの受容性を社会的なものとする実務的なものに分け、実務的な受容性の中に有用性 (usefulness) を位置づけ、さらに、その下に実用性 (utility) とユーザビリティを対比して配置している。実用性は、機能や性能のことであり、他方、ユーザビリティは、学習のしやすさ、使いやすさ、記憶しやすさ、エラーの少なさ、主観的満足度、といったことを指している。これは、ISO/IEC 9126 (Software engineering -

Product quality) [3]のユーザビリティの定義に近いものである。ISO/IEC 9126では、理解のしやすさ、学習のしやすさ、操作のしやすさ、魅力的であることがユーザビリティの属性として挙げられている。

他方、ISO9241-11 (JIS Z 8521) (Ergonomics of human system interaction-Guidance on usability) [4]では、『指定された利用者によって、指定された利用状況の下で、指定された目的を有効に、効率よく、利用者が満足して達成できるための製品の能力』と定義している。これは、ISO/IEC 9126における利用品質 (Quality in Use) と同義である。

ユーザビリティの概念の枠組みは、後者の方が広く、組み込みシステムとの関連を議論する上では、この定義の方が全体の問題を俯瞰する上で有効である。したがって、これ以降では、ユーザビリティを利用品質と同義として取り扱う。それでは、次にこの利用品質とは何であるのか。この定義を咀嚼してみる。

利用品質とは、図1に示すように、ユーザの側から見た品質の概念である。一般的な品質は、製造側が設計目標として、便宜的に設定した目標値として管理される。しかし、ユーザが利用する場合は、この概念が、そのまま使われるわけではない。ユーザにとっては、システム

が使える(ユーザの利用目的に合っている)ものなのか、スムーズに使えるものなのか、そして、使ってよかったと思えるかが重要なことである。このことは、別な視点でいえば、システムを通じて、ユーザが経験することの総体が開発対象となっていることを意味している。この開発対象はユーザエクスペリエンス(ユーザ経験)といわれている。ここには、情緒的な要素も含まれる。

今後、組み込みシステムの開発スコープが、ユビキタスコンピューティングの広まりによって、操作 タスク活動(仕事) 生活様式というように広がっていくことが想像できる。そのため、組み込みシステムにとってのユーザビリティとは、組み込みシステムのソフトウェアとハードウェアを含めた全体とユーザの生活全てに関わるものである。開発フェーズでは、設計にとどまらず、企画、設計、製造、販売などの全てのプロセスに関係するものである。そして、ユーザ側からの利用品質を確保し、向上させるためには、これらまでの開発プロセス全体を強化する必要がある。本稿では、このための方策として、組み込みシステム開発に対するユーザビリティ工学の貢献の可能性を検討する。

2 組み込みシステム開発課題とユーザビリティ工学

2.1 ユーザビリティから見た組み込みシステムの課題

ユーザビリティに関連する組み込みシステムの基本的な問題は、利用におけるトラブルである。最近の話題では、品質トラブルによる回収の問題がある。一般的に、回収コストは企画、設計、製造の段階を経ることによって、10倍ずつ増えるといわれている。効果的なコスト低減は、開発上流工程において適切にユーザビリティを確保することである。また、品質トラブルの延長上にあるのが事故である。バグがない場合でも、想定される利用状況を

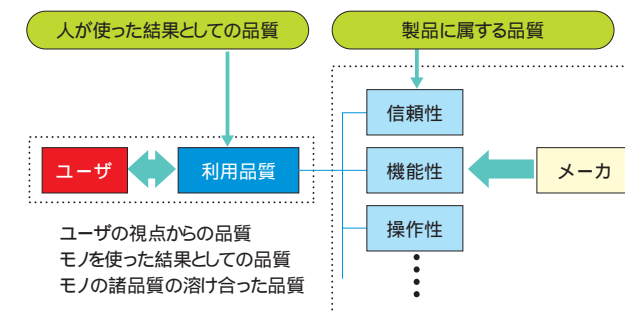


図1 利用品質(ユーザビリティ)の考え方

誤ると事故になる。最悪の場合は、死亡事故につながる。これらは、法的には製造物責任法に関する問題である。

さらに、利用に際する不具合がなくなった場合でも、ユーザにシステムを購入してもらわなければ、経営が成立しない。これは、ブランディングに関わる問題である。品質が安定した後も、使いやすさが悪いとユーザのリテンション効果が得られない。むしろ、二度と使われない危険性をはらむ。システムを通じて、意味あるユーザ経験をしてもらうことを最終的な目標にすると、新たな課題に挑むことになる。図2には、ユーザから見た品質の段階を示した。

以上のように、組み込みシステムにおけるユーザビリティ改善活動の効果は、品質安定による利用時のトラブルを減少化する効果と、新しい利用経験を提案するという2方向の効果がある。

2.2 ユーザビリティ工学とは

実際、品質に関する気づきやよい提案は、設計者の『センス』あるいは、『運』ということ暗黙的に理解されていることが通例である。これらの問題に対して、実践的な技術体系化を進めてきたのが、米国のノーマン[5]、[6]や英国のシャッカール[7]、[8]である。社会的に工学として認知されることへ影響を与えたのが、ニールセンの『ユーザビリティ工学』である。1999年には、ユーザビリティを向上させるための設計プロセスが人間中心設計として体系化された、ISO 13047 (Human-centred design processes for interactive systems) [9]が発行された。

このように、ユーザビリティに関する諸活動は、単なる経験則の集積ではなく、工学の明確な専門分野 (discipline) として確立されつつある。ユーザビリティ工学の技術体系を開発技術と管理技術に分けると、図3のように整理することができる。

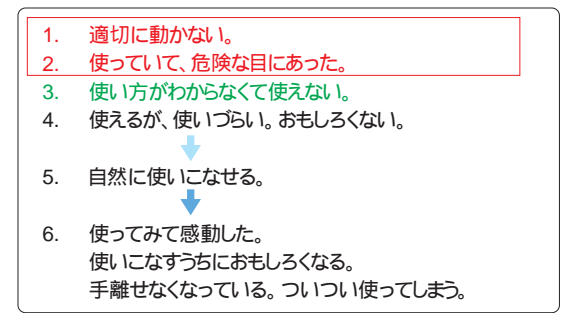


図2 ユーザから見た品質の段階

2.3 組込みシステム開発課題とユーザビリティ工学

では、ユーザビリティ工学が、組込みシステムの利用品質の確保と向上に関連して、開発課題に対してどのような協働と貢献が可能であるのかを以下に示す。組込みシステムの開発課題は、経済産業省「組込みソフトウェア開発力強化推進委員会準備会組込みソフトウェアエンジニアリング部会」の報告書を参考にしている[10]。

(1) プロセス改善・評価

現在、組込み開発プロセスの改善・評価に関して、CMMIやISO15504 (Information technology - Process assessment -) [11]の適用が検討されている。しかしながら、経産省の報告書研究会でも指摘されているように、これらは、情報サービス系のソフトウェアを対象としているために、組込み開発にはそのまま利用できない。ユーザビリティの観点で言えば、ユーザ要求の獲得、記述、検証など、表記されていないプロセスがある。前述の2つのモデルを活用しても、ユーザビリティの向上に結びつかないという報告もある[12]。その原因は、現行モデルには、ユーザビリティに関係するプロセスが考慮されていないためである。そのため、現状のプロセスモデルにユーザビリティ関連プロセスを補完する必要がある。このプロセスに利用できる規格が、ISO13407あるいは、その下位規格である。ISO TR 18529 (Ergonomics of human system interaction-Human-centred lifecycle process descriptions) [13]などである。

ライフサイクルプロセスモデルは、開発管理の基本である。したがって、ユーザビリティの向上を目指すのであれば、人間中心設計プロセスを含めた統合的なプロセスモデルを構築する必要がある。

(2) 開発技術

ユーザビリティ工学は、設計、要求仕様、検証&確認技術に関する技術を蓄積してきた。その中でも、組込みシステムと最も関連のある専門技術は、UI (User

Interface) 設計である。UIに関しては、1980年代後半から、UI設計の経験則や認知心理学の知見から様々なガイドラインが集積されるようになってきている[14]、[15]。また、大手企業のデザイン部門には、専門のUIデザイナーが配置されているのが一般的になっている。コンシューマ市場に直面する組込みシステムは、UIの要素が、ユーザが購買を決定する主たる要因となる。

しかしながら、中小企業の場合、UI設計を担当するのは、通常、システム設計者である。コンシューマ向けシステムのUI設計は細かな仕様が購買の要因となるため、後工程における市場動向の変化やユーザ評価によるフィードバックによって、仕様変更され、手戻りが発生することがある。UI設計は、基本的に『動く』ものを作ることである。しかし、どういうUIが受け入れられるのかがわからなければ、後に仕様変更を余儀なくされる危険性は大きくなる。

これを、UIの差異について、図を元に説明する。図4のUIは、画像をブラウズするものである。左側のブラウザは、クリックするとそのまま次の画像に切り替わるものである。右側は、それにデザイン要素を加えたものである。例えば、クリックする部位に影付けすること、クリックすることによって音などのフィードバックを加えること、画面の遷移がフェードアウトすること等を加えることができる。結果として、明らかに右側のUIのユーザ受容が高まる。ユーザは一度よいUIを経験すると、悪



図4 ユーザインタフェースの違い (赤枠)



図5 仕様決定が困難な場合 (メールの入力中に着信があった場合のインタフェースの例)

	技術系	管理系
企画	ユーザ調査、マーケティング 要求分析・定義 要求仕様の検証	ユーザビリティ・プロジェクトマネジメント ヒューマンリソースマネジメント 品質マネジメント
設計	概念設計 UI設計 マニュアル設計 Web設計 ビジュアルインタフェース プロトタイプ	リスクマネジメント(PL等) 知財マネジメント(UI特許など) 技術管理(ツール、手技法の管理)
評価	ユーザビリティ・インスペクション ユーザビリティ・テスト 長期モニター評価	
サービス運用	サービス・運用設計	

図3 ユーザビリティ工学の技術

いUIを利用することへ強い拒否感を持つ傾向がある。そのため、UIは組込みシステムの品質を決定する重要な要因である。このプロセスにおける主なユーザビリティエンジニアの仕事は、UI設計を支援するか、設計者のUI設計を支援するUI設計ガイドラインを編集することである。専門のユーザビリティエンジニアがいない場合は、個々の組込み設計者がUI設計を担うことになる。広く、一般の設計者が研修できる教育訓練環境を整備することが求められる。

一方、UIの良否は、一意的に決定できないこともある。図5は、携帯でメールを打っているときに、電話が割り込んできた場合の例を示している。このような場合、電話を保留するのか、メールを保存して電話を受けるのかなどの仕様は一意的に決定することができない。上位の要求仕様定義プロセスにおいて、どういう利用者が、どういう目的で、どういう利用状況で利用するかが明確にされていないと決定できない。これは、要求定義・獲得フェーズの課題である。ユーザビリティ工学には、ユーザ分析に関する様々な手技法の蓄積がある。さらに、得られたデータから、要求事項を抽出し、それを記述し、検証することは、ユーザビリティ工学の主要な技術である。特に欧米では、様々な研究開発がなされてきた[16]、[17]。しかしながら、日本語で入手できる、組込みシステム開発におけるこの領域の研究資料は極めて少ない[18]、[19]。

組込みシステムの手戻りの大きな要因として、要求仕様の不備が指摘されている[20]。このことを考えると、このフェーズに対してユーザビリティ工学が貢献できる可能性は高い。

最後に組込みシステム開発の検証・テストフェーズへの技術として、ユーザビリティ評価を挙げることができる。ユーザビリティ評価は、ユーザの側に立って、そのシステムが利用できるのか、そして利用できる場合には、効率的に使えるのか等を分析するものである。主なものには、ユーザが利用する状況を分析し評価するユーザビリティテストと、専門家が仕様等から評価するインスペクションがある。この評価活動は、設計フェーズから、最終の引き渡しのフェーズまでのどの段階でも必要である。特に、確認 (Validation) には、実際のユーザによるユーザビリティ評価が果たす役割は重要である。国内では、ユーザビリティ評価は、徐々に浸透しつつある[21]、[22]。

以上のように、ユーザから見た様々な検証および確認プロセスにユーザビリティ工学は欠かすことはできないものである。

(3) 管理技術

前述のように、組込みシステム開発において、要求定義・獲得技術、UI設計技術、ユーザビリティ評価技術等を十分に機能させるためには、これらが適切に管理されるように運用される必要がある。組込みシステムの開発管理課題には、開発計画立案手法に人間中心設計プロセス管理を加えること、UI設計ガイドラインやスタンダードの管理、ユーザビリティ工学の手技法の管理などを加えることによって強化される。

品質マネジメントについては、ユーザビリティや利用品質のメトリックスを管理すること、ユーザビリティ・インスペクションやユーザビリティテストの手法や運用を管理すること、そして、評価結果から改善への反映方法を管理することなどを挙げることができる。

さらに、ユーザビリティ工学の特徴から、製造物責任問題のように、ユーザに与えるリスクを管理することへの貢献は大きいといえる。また、要員管理の上では、組込みシステムのスキルスタンダードにユーザビリティスキルを統合し、さらに高いレベルや多方面のスキルを包含した、ハイブリッドなスタンダードを構築することが可能である。

以上、組込みシステムの開発課題に対する貢献を表1にまとめた。これを、組込み業界の企業関連を考慮して、ユーザビリティ専門家の活動を展開したのが図6である。

3 人間中心設計プロセス研究

利用品質の確保あるいは改善は、人間中心設計プロセスを実装することによって遂行される[23]。前述のように、ISO 13407は、基本的な人間中心設計プロセスを規定した規格である。また、プロセスを詳細化したプラクティスについては、ISO 18529にて参照可能である。人間中心設計プロセスは、ユーザビリティの開発技術や管理技術を統合的に管理する上での基盤となるものである。したがってユーザビリティの組織的な強化のためには、人間中心設計プロセスの管理は必須事項である。

以下は、筆者がこれまで従事してきた人間中心設計プロセス研究に関する事例の紹介をしながら、利用品質を組織的かつ持続的にマネジメントする方法を検討していく。

表1 組み込みシステム開発におけるユーザビリティ工学の貢献

カテゴリ	サブカテゴリ	ユーザビリティ工学による貢献内容
P:プロセス評価改善		・人間中心設計プロセスを統合した組み込みソフトウェア向けプロセス定義の拡張
		・人間中心設計プロセスを統合した組み込みソフトウェア向けプロセス評価手法の強化
		・人間中心設計プロセスを統合した組み込みソフトウェア向け改善策とリンクしたプロセス改善手法の強化
		・人間中心設計プロセスを統合した組み込みソフトウェア向けプロセス評価改善のための運用の仕組み整備
TS:開発技術	RE:要求定義・獲得フェーズ	・組み込みソフトウェア向け要求獲得手法の強化 ・組み込みソフトウェア向け要求の分析と定式化手法の強化 ・組み込みソフトウェア向け要求仕様の活用技術の強化
	DM:設計フェーズ	・UI設計モデリング技術 ・UI設計可視化技術 ・UI設計検証技術 ・UI設計再利用技術
	IT:実装フェーズ	(特定の支援なし)
	V&V:検証・テストフェーズ	・ユーザビリティテスト実行環境の整備 ・ユーザビリティテスト項目作成と優先度付け ・ユーザビリティ・インスペクション手法の強化 ・ユーザビリティテストの効率化
	MS:管理技術	MT:開発管理
	QM:品質可視化	・ユーザビリティ、利用品質メトリクスの設定と計測法 ・ユーザビリティ・インスペクションの強化と普及 ・ユーザビリティ評価結果の反映方法

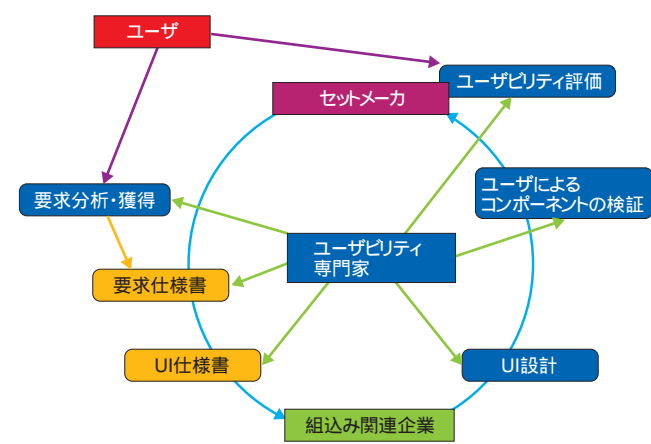


図6 組み込みシステム関連企業におけるユーザビリティ専門家の役割

(1) 組み込みシステム用人間中心設計プロセス成熟度モデルの構築

ISO 13407に記述された人間中心設計プロセスは、様々な開発ドメインに共通のユーザビリティ活動を集約したものである。そのため、抽象度が高い。実際の開発ライフサイクルプロセスは、開発対象によって異なるため、ユーザビリティに関するプロセスを管理するためには、それぞれのものづくりのプロセスと統合する必要がある。図7は、一般的なものづくりプロセスに対して、人間中心設計プロセスを展開した事例である。

次に、各々のプロセスの遂行能力を評定する必要がある。これは、プロセスの能力成熟度 (Capability Maturity) である。結果として、組み込みシステムの作り込みに応じ

た、ユーザビリティを向上、改善するプロセスは、図8のように、組み込み開発ライフサイクルプロセス×人間中心設計プロセス×プロセス成熟度の3軸から構成されることになる。

一般的な設計プロセスの能力成熟度モデルについては、この3軸モデルの考え方で構築することが可能である。しかしながら、組み込みシステムの中には、コンシューマ向け製品が多い。そのため、企画提案するシステムが、どのようなユーザが受容し、どのような状況で利用し、その状況にどのような影響を与えるのかを、予め予測できるのであれば、都合がよい。結果として、開発における手戻りのリスクを軽減でき、さらには、市場へ影響を与えるシステムを創出する可能性を広げることができる。また、システム構想は、事業戦略に基づいて、市場戦略、技術戦略から導かれることになっているが、実際は、市場を意識しながらも、技術的な『ジャスト・アイデア』に頼る場合が少なくない。事業戦略に対して適切な構想プロセスを踏んでいるかを評価することができれば、その効用は大きい。

筆者らが開発した人間中心設計プロセス成熟度モデル (HCD-PCM: Human-centered Design Processes Maturity Model) は、この構想プロセスと設計プロセスを統合したハイブリッドなモデルである。モデルの全体構想を図9に示した。

このようなモデルは、ユーザビリティ能力成熟度モデルと呼ばれるものである。既に、いくつかのモデルが提

案されている。2003年から国際的なユーザビリティ研究者との協働によって、公開されているユーザビリティ能力成熟度モデルを調査した結果、十数種のモデルを特定することができた[24]。これを表2に示した。さらに、これらモデルと他の品質モデルとの相関図を示したものが、図10である。この図から、HCD-PCMがユニークな位置にあることがわかる。

人間中心設計プロセス	
Requirements Definition & Analysis	利用状況の分析 ユーザ要求の定義 ユーザ要求の分析 利用品質計画 長期モニタリングの計画
Design	ヒューマンインタフェース設計 モックアップによる検証
Implementation	プロトタイプ製作 システム運用方法の開発 保守およびサービスシステムの設計
Integration	統合化システム設計 統合化システム運用方法の開発
Verification & Validation	システム構成要素の検証 システムの確認
Operation & Maintenance	システムの導入 システムの保守
Disposal	システムの回収

図7 ものづくりプロセスへの人間中心設計プロセスのマッピング

(2) プロセス成熟度モデルの活用

開発された成熟度モデルは、いくつかのプロジェクト事例を基に、各事業におけるプロセスを評価することに活用される。評価は、インタビューや資料をもとに行われるが、形式的に行われた場合、調査対象者が、実際のプロセスではなく、理想的なプロセスに基づいて回答したり、忘れていていることがある。実際に、開発プロセスは、暗黙的に遂行されることが多いため、この暗黙的なプロセスを一度、意識化させて、相対的に俯瞰する必

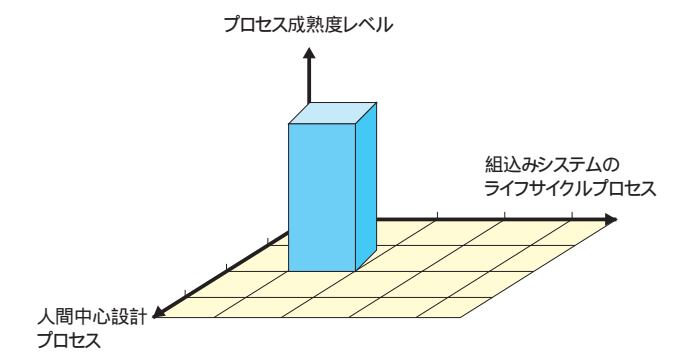


図8 組み込みシステム版人間中心設計プロセス成熟度モデル

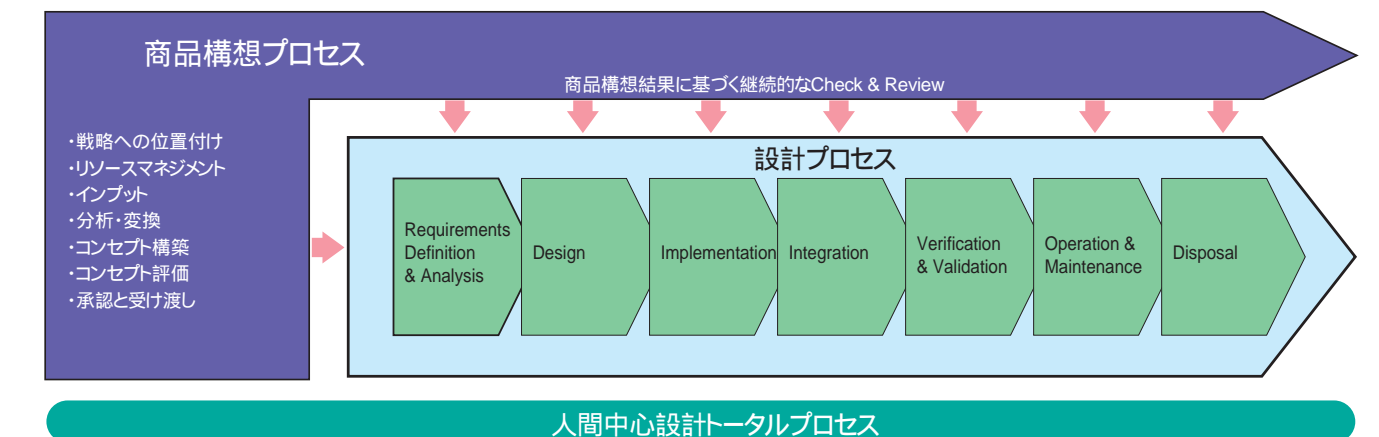


図9 人間中心設計プロセス成熟度モデルの全体像

表2 ユーザビリティ能力成熟度モデルのリスト

略語	モデル名(開発者)
Trillium	Trillium(Bell Canada)
ULMM	Usability Leadership Maturity Mode(IBM)
HPA	Humanware Process Assessment(Philips)
UCDM	User Centred Design Maturity(the Ergonomics and Safety Research Institute, ESRI)
UMM-HCS	Usability Maturity Model: Human-Centredness Scale(European INUSE project)
UMM-P	Usability Maturity Model: Processes(European INUSE project)
ISO 18529	ISO 18529 Human-Centred Lifecycle Process Descriptions
KESSU	KESSU UCD performance assessment(Oulu University (Finland))
DATech-UEPA	Procedures for Usability Engineering Process Assessment(DATech (Germany))
HCD-PCM	Human-centered Design Processes Capability Maturity Model (三菱総研, NTT-AT, 小樽商大)
HFIPRA	Human Factors Integration Process Risk Assessment(UK Government HMIS and HFICMM projects)
ISO 18152	ISO 18152 A specification for the process assessment of human-system issues, + ISO 15504 Process Assessment

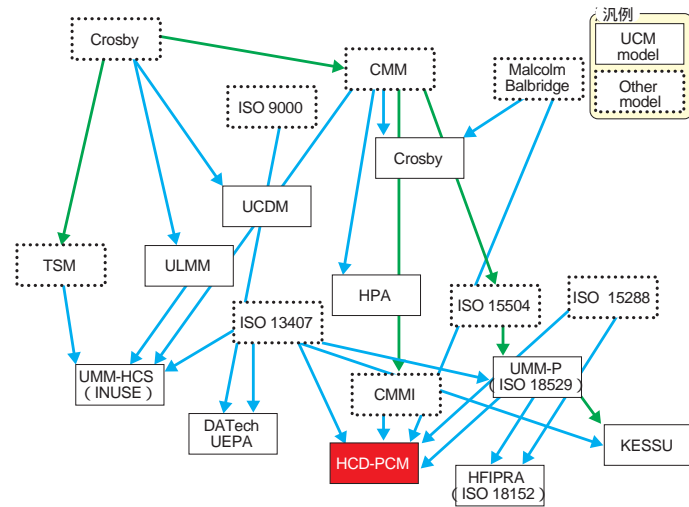


図10 ユーザビリティ能力成熟度モデルの相関図



写真1 プロセスアセスメントの実施風景

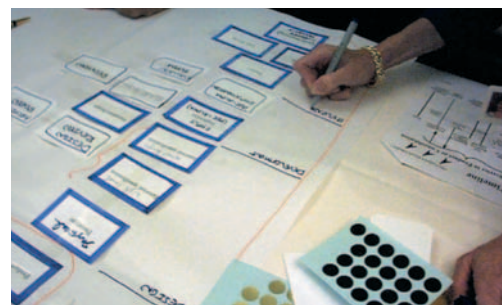


写真2 プロセスカードの配置とプロセス定義

製造業では大手企業でも整備されているわけではなく、まして中小企業では、認知されていない可能性が大きい。実際に、札幌市のIT企業を調査した結果、人間中心設計を導入している中小企業は皆無であった。このような状況の中で、プロセス評価を型どおりに実施した場合、設計者には決してよい印象を与えず、かえって動機づけを悪くする結果になる。むしろ、アセスメントよりもワークショップ形式で、実際のプロジェクト計画に立案しながら人間中心設計プロセスを学習していく方が望ましい。PROFES 2004におけるチュートリアルでは、具体的なシステム開発プロジェクトを計画する課題を具体的に考える中で、必要となるプロセスが何であるかを発見的に研修する方法を実施している。

5 まとめ

本稿では、ユーザビリティ工学が目指す品質概念として、利用品質という考え方を説明した。利用品質の向上は、今後の組み込みシステムの市場の発展を考える上で、欠かせない指針と思われる。この利用品質という高いレベルの品質基準をクリアするためには、現行の開発プロセスに人間中心設計プロセスを統合し、強化あるいは補完する必要がある。実際に、組み込みシステム開発が直面する課題とユーザビリティ工学の技術とを比較すると、それらの課題に貢献する可能性が大きいこともわかった。すなわち、今後、我が国の組み込みシステム開発が競争力を高め、市場を創出する力を高めるには、ユーザビリティ工学の適用は、期待できるものである。組み込みシステム開発とユーザビリティ工学の協働によって、新しいものづくりの形が生まれる可能性がある。

一方、ユーザビリティ工学は、まだ新しい研究領域でもある。そのため、大学等の高等教育機関では、体系立って教育が行われる体制はできていない。そのため、設計者が新たに教育訓練を希望してもなかなか対応できない現状にある。段階を追ったスキルレベルを構築することによって、まずは、基本的なスキルを習得できる環境を整備することが急務である。

最後に、最近の国内のユーザビリティ活動の動向に触れておきたい。まず、文部科学省の知的クラスタ創生事業における『札幌ITカロッツェリア』創生事業がある[28]。これは、ユーザ要求を適切に把握し、利用品質の高い組み込みシステムのプロトタイプを、効率的に開発する

ことを目的としている。例えば、何か技術的なアイデアがクライアントにある場合、それを基にしてユーザと協働しながら、コンセプト構築から要求分析・獲得を経て、具体的な要求仕様を作成することを支援できる。さらには、要求仕様に基づいて、UIおよび組み込みソフトウェアを設計し、同時に高機能なCADを利用して筐体をデザインし、光造形による、動作可能なプロトタイプをスピーディに開発する。開発したプロトタイプは、ユーザビリティ評価を経て、改善され、クライアントへ渡すことになる。個々のユーザビリティ活動を組み込みソフトウェア、次世代CADを活用したデザインと協調した、新しい設計プロセスである。ユーザビリティ関連プロジェクトに大規模な国家予算が投入されるのは初めてのケースである。

また、本年2月19日をもって、ユーザビリティ活動を普及するための法人『人間中心設計推進機構』がNPO法人として認承された[29]。このNPO法人には、ほとんどの国内のユーザビリティ専門家が参集している。研究、教育を土台として、人間中心設計の社会普及活動、受託業務、規格・認定/認証活動を事業とすることを唱っている。特に、教育機関において、ユーザビリティ関連の教育訓練が整備されていない現状では、人材育成の役割が強く期待される(図11)。

日本の場合、ユーザビリティへの国際的な動向に対して、10年以上の開きがあるのも事実である。組み込みシステムのユーザビリティ先進例としては、欧州のNOKIAやPhilipsなどの製造業があり、ユーザビリティが製品開発へ強い影響を与えている。コンカレントなものづくりと

- 1.人間中心設計に関する研究活動
ユーザビリティを探索する調査・研究事業を受託
- 2.人間中心設計に関する教育活動
ユーザビリティノウハウに関する講演会、教育セミナー事業を実施
- 3.人間中心設計の社会化活動
ユーザビリティを向上させる人間中心設計の普及・啓発事業
人間中心設計の大切さ・必要性を世の中の人に認知してもらう
商品やサービスのユーザビリティに関するコンサルテーション事業
ユーザビリティの向上を考える人材のネットワーク形成事業
ユーザビリティ評価、ユーザ調査分析に積極的に協力するモニターを組織する活動
ユーザビリティの良い商品を広く一般消費者に伝える広報・販売活動
年間Awards審査、発表など
- 4.人間中心設計の業務活動
大規模なプロジェクトや複雑なプロジェクトの業務を受託する
公共性の強い社会基盤システムのユーザビリティの検討、設計活動
ユーザビリティのよい商品・サービスの設計・開発並びにその人材育成事業
ユーザビリティに関する情報収集並びに情報提供事業
全産業から生み出される商品・サービスのユーザビリティを評価・分析する事業
- 5.人間中心設計に関する規格化/認定活動
ユーザビリティの判定に有効な指標の規格化および認証事業
ユーザビリティ人材およびユーザビリティ商品等の認証活動

図11 NPO法人『人間中心設計推進機構』の事業概要

して特徴づけられる我が国の開発に対して、ユーザと協調することを補完して、どのようにして利用品質の高い製品を創出してゆくのか。利用者側に立った新しいものづくりの水準に向かってコンセンサスを形成し、まずは、ユーザビリティの国際競争に向けて、スタートラインに着くことが急務であると考える。

参考文献

- [1] 黒須正明他：特集 ユーザビリティ・エンジニアリング、情報処理、Vol.44, No.2, 2003
- [2] Nielsen, J. (篠原稔和監訳)：ユーザビリティエンジニアリング原論、株式会社トッパン、1999
- [3] ISO/IEC 9216, Software engineering - Product quality 2001
- [4] ISO 9241-11 (JIS Z 8521), Ergonomics of human system interaction_ Guidance on usability (人間工学 - 視覚表示装置を用いるオフィス作業 - 使用性についての手引き), 1999
- [5] Norman, D., Draper, S.: User Centered System Design, Lawrence Erlbaum Associates, 1986
- [6] Norman, D.: 誰のためのデザイン?, 新曜社, 1990
- [7] Shackel, B.: Ergonomics in information technology in Europe -a review, BEHAVIOUR AND INFORMATION TECHNOLOGY, 4, 4, 263-287, 1985
- [8] Shackel, B.: Richardson, S., Human factors for Informatics Usability - Background and Overview. In Shackel, B and Richardson, S (ed.), Human Factors for Informatics Usability, Cambridge University Press, 1991
- [9] ISO13407 (JIS Z 8530), Human-centred Design Processes for Interactive Systems, 1999
- [10] 経済産業省組み込みソフトウェア開発力強化推進委員会準備会組み込みソフトウェアエンジニアリング部会：組み込みソフトウェアの開発力向上に向けた施策と提言、経済産業省、2004
- [11] ISO/IEC 15504, Information technology - Process assessment -, 2004
- [12] Jokela, T., Lalli, T.: (2003) Usability and CMMI: Does A higher Maturity Level in Product Development Mean Better Usability? CHI 2003, ACM, 1010-1011, 2003
- [13] ISO/TR 18529 Ergonomics of human system interaction _Human-centred lifecycle process descriptions, 2000
- [14] ベン・シュナイダーマン(東基衛, 井開治(翻訳))：ユーザー・インタフェースの設計 やさしい対話型システムへの指針、日経BP社、1995
- [15] アラン・クーバー(テクニカルコア(翻訳))：ユーザーインターフェイスデザイン Windows95時代のソフトウェアデザインを考える、翔泳社、1996
- [16] Newman, W.他(北島宗雄監訳)：インタラクティブシステムデザイン、ピアソン・エデュケーション、1999
- [17] Beyer, H., Holtzblatt, K.: Contextual Design, Morgan Kaufman, 1998
- [18] 平沢尚毅：組み込みシステムの要求仕様定義、システム設計編(Dコース)、(株)半導体理工学研究センター、2004
- [19] スザンヌ・ロバートソン、ジェームズ・ロバートソン(苅部英司訳)：要件プロセス完全修得法
- [20] 経済産業省 商務情報政策局 情報政策ユニット 情報処理振興課：2004年版組み込みソフトウェア産業実態調査報告書、監修：組み込みソフトウェア開発力強化推進委員会、2004
- [21] 黒須正明他：ユーザビリティテスト、共立出版、2003
- [22] Dumas, J., Redish, J.: A Practical Guide to Usability Testing, Intellect L & D E F a E, 1999
- [23] 黒須正明他：ISO13407がわかる本、オーム社、2001
- [24] Jokela, T., Siponen, M., Hirasawa, N., Earthy, J.: A Survey of Usability Capability Maturity Models: Implications for Practice and Research, BEHAVIOUR AND INFORMATION TECHNOLOGY, (受理済)
- [25] Kurosu, M., Itoh, M., Horibe, M., Hirasawa, N., Fujiwara, Y.: Diagnosis of Human-Centeredness of the Design Process by the SDOS, Proceedings of Usability Professional Associations, 2000
- [26] 平沢尚毅 他：人間中心設計の実現を目指した設計プロセス改善手法(2), 人間工学, Vol.36, 特別号, 2000
- [27] 人間生活工学センター(HQL), 人間中心設計に係わる国際規格への対応に関する調査研究, 2000
- [28] <http://www.it-cluster.jp/>
- [29] <http://www.hcdnet.org/>

海外レポート

鶴保 征城

去る2月27日から3月5日にかけて独、英、アイルランドを駆け足でまわってきた。4機関を訪問し、ソフトウェア工学関係の研究動向について多くの知見を得ることができた。IESEでは現在実施している共同研究の成果の確認、次年度プロジェクトの確定、及びそれ以降の共同研究テーマの可能性について議論した。

1. ドイツ IESE

IESE⁽¹⁾はソフトウェア工学に関する基礎研究と産業界での応用との間のギャップを埋め、技術移転を進める役割を果たしている。基礎研究側にMax-Planck研究所とKaiserslautern大学、産業側に多くのコンサルタント企業やソフトウェア開発企業が存在する。SECや米国SEI⁽²⁾も同じような産学連携の役割を果たすものであるが、三者を比較すると図のような違いがある。SECは現在産業寄りのスタンスで実践的な取り組みを行い、産業界へのソフトウェアエンジニアリングの普及を図ろうとしているが、今後はカバー範囲を拡大するか、または外部組織と戦略的連携を図ってギャップを埋める必要があると思われる。

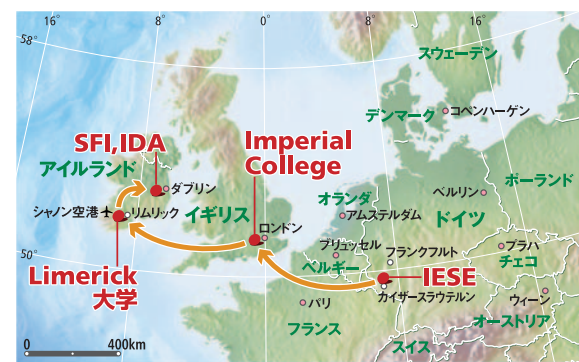
Rombach所長とはSECとIESEの共同研究が順調に進展していることを確認し、さらに今後の共同研究テーマとして次の分野を検討することで合意した。

定量的なモデリングと分析

プロセスアセスメント（特に、自動車分野等のドメイン毎の動きを念頭において）

プロセス改善やプロセスモデリング

プロジェクトにおける計測及びコントロール



Kaiserslautern市はFrankfurtからICE（InterCity Expresszug）という高速鉄道で南へ約1時間半のところにある。奇縁であるが、IPAが所在する文京区とは姉妹都市の関係にある。驚いたのは寒いことだった。我々の前週に滞在していたSECの菊地研究員から寒いとは聞いていたが、タクシーの温度計がマイナス16度を指していたのにはびっくり。

2. イギリス Imperial College

面談したJeff Kramer教授は要求工学の権威である。教授は要求工学とアーキテクチャの研究を同時に進めることが必要と考え、両者を統合したアプローチを行っている。今秋には日本の研究者との共同研究（ワークショップ）も計画されている。研究の例として、自己組織化ソフトウェアシステムに関する研究がある。ソフトウェアは時間が経過するにつれて進化していくものであると仮定し、このダイナミックな変化に対応する開発およびコンポーネント化を実現しようとしている。この他に、ソフトウェア開発における分析・設計のためのモデル化手法、およびモデルそのものの提供を目指したシステムモデリングの研究、モバイル環境や家電ネットワーク等に対応したコンピューティン

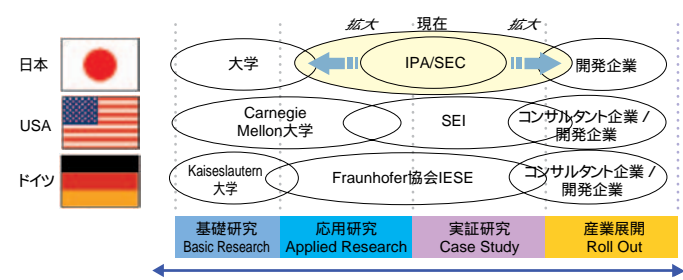


図1 各国のソフトウェア工学関連組織とその役割

グ環境や効果的な開発を目指したユビキタス・コンピューティングの研究が実施されている。

Kramer教授が属しているDepartment of Computingの入学希望者は、かつて15倍程度だった倍率が5倍程度に落ちているとのことであった。後述のLimerick大学では定員割れしているそうであり、2000年前後のITバブルの崩壊以降、日米欧ともにコンピュータ関係の人氣が低下していることがうかがえる。

3. アイルランド Limerick大学, SFI/IDA

アイルランドは、J.F.Kennedyの故郷、三つ葉のシャムロック、トム・クルーズとニコール・キッドマンの映画「遥かなる大地へ」の舞台などで知られているが、ソフトウェア産業を積極的に育成していることでも有名である。面積が7万平方キロ、人口が400万人弱で、北海道（面積8.3万平方キロ、570万人）よりも若干小さな小国。1990年代に入ってからソフトウェア、医薬、エレクトロニクス等のハイテク産業の育成に的を絞ることによって、日本が低成長に悩まされた1990年から2003年にかけて実に3.8倍ものGDP成長を達成した。特にソフトウェア製品の輸出は1998年に33億USドルを達成しアメリカを抜いて世界一になっている。ちなみにこの年の日本のソフトウェア輸出はわずか1.7億USドルに過ぎない。

アイルランドの4大総合大学の1つであるLimerick大学では今回の訪問をアレンジしていただいたKevin Ryan教授がISERC⁽³⁾を率いている。このミッションは、アイルランドのソフトウェア産業の競争力強化にある。かなり学寄りのスタンスでありISERCの役割をリサーチとパイロットまでとしている。一方、アプリケーションドメインをAutomotive、Telecom、Medical、Financeに設定することによって、産業界への技術移転の効率化を狙っているようである。Ryan教授から連携の話があったので現在詳細を詰めている。

表1 訪問先

訪問日	訪問機関
2月27日	ドイツフ라운ホーファー協会IESE(Institute Experimentalle Software Engineering)
3月1日	Imperial College (Department of Computing)
3月2日	ISERC (Irish Software Engineering Research Consortium)
3月3日	SFI (Software Foundation Ireland) IDA (Investment and Development Agency)

Limerick大学での収穫の1つにDavid L.Parnas教授に会えたことがあげられる。Parnas教授は、Dijkstraの構造化と並ぶソフトウェア工学の重要な概念である情報隠蔽の提案を1972年に発表している。現在の研究テーマはテーブル記法を使った要求定義・網羅性の確保であり、ソフトウェア工学に関する種々の意見を聞くことが出来た。Limerick大学とは5年間の契約とのことであった。

DublinではSFI⁽⁴⁾とIDA⁽⁵⁾を訪問した。紙数の都合で説明は省略するが、アイルランドが1970年代の製造業誘致の失敗後、1990年代に入ってサービスセクターになるという戦略転換により見事に躍進したことは日本にとって大いに参考になるものと思われる。



アイルランドLimerick大学のゲストハウスにて
写真左よりVincent Cunnane副学長、新谷勝利SEC研究員、David Parnas教授、鶴保征城SEC所長、Kevin Ryan教授、石谷靖SEC研究員

(1) IESE : Institute of Experimental Software Engineering. イエゼと読む。 <http://www.iese.fhg.de/>

(2) SEI : Software Engineering Institute <http://www.sei.cmu.edu/sei-home.html>

(3) ISERC : Irish Software Engineering Research Consortium <http://www.iserc.ie/people/universityoflimerick.html>

(4) SFI : Science Foundation Ireland <http://www.sfi.ie/home/index.asp>

(5) IDA : Investment and Development Agency <http://www.idaireland.com/home/>

ソースコードの品質向上技術

ソフトウェア・エンジニアリング・センター プロジェクト統括グループ
研究員

川井 奈央

ここでは、年々複雑化、短納期化するソフトウェアの品質確保のために使用されるコーディング規約に関して紹介する。また、その利用方法や、課題についても紹介し、今後のソフトウェアの品質向上に役立ててもらいたい。

1. 背景

組込みソフトウェア開発は、ハードウェアに依存する部分なども多いため、未だにC言語等を使用して実機のソースコードを書く機会が多い。ソースコードは最終的なソフトウェアとして提供する機能や特性そのものであり、その品質は極めて重要視されなければならないものである。また、近年では、機能が多様化、複雑化し、そのソースコードも膨大な量になっている。このような膨大な量のソースコードを、限られた時間内で如何に確認し、その品質を確認・確保していくかが大きな問題となっている。例えば、自動車に搭載される制御装置等車載系のソフトウェアは、ソースコード上の1つのミスにより数十億の損害を出してしまうといった重大な損失に直結するものも少なくない。欧州では、このような車載ソフトウェアのソースコード上のミスを防ぐために、車載系ソフトウェア向けのコーディング規約 (MISRA-C) を整備し、ミスを極力減らす努力がなされている。

本解説では、組込みソフトウェアのソースコード品質について、実際の事例を考察し、またソースコード品質向上のためのツールの1つであるコーディング規約に着目し、そのツールとしての特色や課題等を紹介する。

2. ソースコードの品質

2.1 品質特性

ソフトウェアの品質という、その代表的な要素として不具合の有無が語られることが多い。しかし、ソフトウェアの品質については、必ずしも不具合の有無だけで評価することは好ましくない。例えば、携帯電話等のソフトウェアでは、何世代にもわたって過去の製品ソフトウェアを利用し続けることがほとんどである。こうした開発形態をとる場合、MPUやOS等の動作環境が変更になった場合に備えソフトウェアが移植しやすい作りになっているか (移植性) また再利用しやすい形で既存のソフトウェア資産が作られているか (再利用性) 等もソフトウェアの品質としては極めて重要な側面となる。このようにソフトウェアの品質は単なる不具合の有無だけでなく、再利用しやすさ、移植のしやすさ等様々な側面から考える必要がある。

こうしたソフトウェア品質の様々な側面を整理したモデルとしてISO/IEC 9126が定められている。ISO/IEC 9126では、製品としてのソフトウェアの品質を考える側面として、機能性、信頼性、使用性、効率性、保守性、移植性の6つの視点と、それぞれを更に詳細に規定する品質副特性を定めている。ISO/IEC 9126はソフトウェアプロダクトの品質を考える視点を提示したものであるが、これらの品質特性を製品として実現するためには、その開発成果物のひとつひとつを、これらの視点で考え、品質

を作りこんでいくことが求められる。これは本稿で考えるソースコードについても同様であり、ソースコード品質についてもISO/IEC 9126で示された視点で考え、その品質を作り込んでいくことが必要となる。

2.2 ソースコードの保守性とコードクローン

ここでは、保守性の視点の事例として、コードクローンを取り上げる。

(1) コードクローンとは

近年、複雑化する組込みソフトウェアの開発では、製品リリース後に発生する保守作業や、機能改良等のために既存ソースコードに手を加えるといった作業が多く発生している。このような場合、対象となるソースコードの保守性の良し悪しによって、保守作業の手間は大きく異なってくる。

ソフトウェアの保守や再利用を行う場合、既存のコードのコピー/ペーストを利用する 경우가少なくないが、こうした方法はソースコードの保守性の面からは必ずしもよい結果を生まないとされている。安易なソースコードのコピー&ペーストの結果生まれるソースコード内の構造が類似した部分 (主にコピー、ペーストによって作られる) をコードクローン (Code Clone) と呼ぶ。

(2) コードクローンの弊害

コードクローンがコード中に存在すると、処理モジュールで不具合等が検出され修正を施した場合、これをコピーして作った処理モジュールの修正変更等も検討しなくてはならない。このようにクローンが内在するコードでは、クローンの数だけ確認、修正しなければならない箇所が増える形となり (修正箇所の拡散) 結果的にソースコードの保守性は大幅に低下することになる。また、このようにして拡散したすべてのクローンを確実に把握し対応できれば問題の発生は少なく済むが、こうしたクローンの存在を把握しきれないような複雑かつ大規模なコードの場合、おおよとのコードに加えた修正がクローンにまで及ばず、結果的に不具合を発生させてしまうといった事態も発生しかねず、ソフトウェアの信頼性にも深刻な影響を及ぼしかねない。

次節以降、ソースコード品質向上のための道具としてのコーディング規約について紹介してみたい。

3. コーディング規約の概要

3.1 コーディング規約とは

コーディング規約、コーディングルール、コーディングスタイル、コーディング作法 ソースコードを取り巻くこれらの規約やルールは、これまで数多くのものが提案されてきた。「規約」「規定」「ルール」「作法」等名前の差こそあれ、いずれもソースコードを記述する際にどのような点に注意すべきかを整理したノウハウ集と捉えることができる (以下では総称してコーディング規約と記す)。コーディング規約の多くは、過去にソースコードの誤った記述によって深刻な品質トラブルを出した事例を参考に、「このような書き方をするとトラブルを引き起こす」という経験を、ルールや規約の形に整理したものである。また、コーディング規約では、このように基本的に企業内の経験に基づいて決められたルール以外にも、ドメインや製品の特性に依存するルールが設けられている場合もある。これらのコーディング規約を守らなかった場合、ソースコード上の不具合を引き起こす場合が少なくないと考えられる。

3.2 コーディング規約の例

図1にコーディング規約の一例を紹介する。図1のコーディング規約では、過去の自部門の経験を参考に再利用性、保守性、移植性等を考慮して整備された規約の一部である。例えば、インデントや括弧、コーディングのスタイルに関する規約はソースコードの保守性や理解し

コーディング規約	説明
難しい式は分けて書く	再利用可能なコードになっているか注意して使う
インデントはわかりやすく書く	他の人が見ても理解しやすいコードになっているか注意しよう
カッコを上手く使ってコードを明確に書く	レビューしやすいコードになっているか注意しよう
goto文は使用しないように書く	メンテナンス性の低いコードになっているか注意して使う
switch文のcase節にはbreakを書く	switch文を使う時に、breakはあるかなどに気をつけて使う
無限ループの使い方には気を付けてコードを書く	必要な部分に無限ループを使用しているのか、本当に必要かを考えて使う
...	...

図1 コーディング規約の例

やすさを意識し第三者がレビューしやすいコードを作成するためのルールである。また、goto文やswitch文に関するルールは、ソースコードの保守性や信頼性を向上させることを目標としている。

3.3 コーディング規約の利用法

このようなコーディング規約を実際の開発現場で利用する場合、その利用方法として次の二つを考慮することができる。

(1) 技術者がソースコードを書く際の参考として利用

技術者は、所属する組織で定めたコーディング規約を参照することで、その組織に適したソースコードの書き方ができるようになり、ソースコード作成時の個人差を少なくすることができる。また、ソフトウェアの不具合につながる危険なソースコードの書き方を未然に防ぐことも可能になり、結果としてソースコードの信頼性や保守性が向上するといった効果が期待できる。

(2) ソースコードレビュー時の基準として利用

コードレビューやインスペクションを行う際には一般的にレビュー基準が必要になる。特に規模の大きなソフトウェアでは、レビューの際に、ソースコードのどこを見るべきか、どこをどのような観点でチェックすべきかといった視点を予め準備しておく必要がある。コーディング規約をこうしたソースコードレビューを行う際の基準として利用することも有効な利用方法の1つである。レビュー基準としてコーディング規約を利用することで、ソースコード品質の安定、向上を実現することも可能である。

4. コーディング規約の整備状況

ここでは、有用な道具としてのコーディング規約が実際の開発の現場で、どの程度整備され、どの程度活用されているかを調査データをもとに紹介してみたい。

4.1 コーディング規約の整備状況

図2、3は2004年に経済産業省が実施した組込みソフトウェア産業実態調査の結果から、コーディング規約の整備状況を抜粋したものである。この図からもわかるように、欧州では90%の企業が、米国では70%の企業が社内

にコーディング規約が存在すると回答している。欧州の自動車業界では、MISRA-C等のコーディング規約がすでに整備され、実際の組み込みシステムの開発現場で使用されている。

一方、我が国では、コーディング規約が存在すると回答している企業は50%であり、コーディングミスが開発手戻りの原因となっていると回答したのは20%ぐらいの企業であった。しかし、この20%は一見少ないようにも見えるが、組み込みソフトウェアでは1つの不具合が数十億の損害を生むこともめずらしいことではなく、この20%をなくすることができるかどうかは品質面で大きな違いがある。実際に組み込み機器の不具合改修、リコール等の損害事例が多々報告されているが、そのうちのいくつかには、こうしたコーディングの誤りを直接の原因とするものも含まれていると考えられる。

4.2 世の中のコーディング規約

現在、世の中には数多くのコーディング規約が提唱されている。国際規格のような位置付けのコーディング規約、自動車のような特定の製品やドメインを対象にしたコーディング規約、あるいは、C++等の特定言語利用者を対象にしたコーディング規約等様々である。また、個々の会社あるいは部門毎のコーディング規約が策定されている場合も少なくない。現在、SECではこうした世の中にあるコーディング規約の系統的な整理を進めている。図4は、この作業の途中経過をまとめたもので、現在、提唱されているコーディング規約の代表的なものの特徴を整理している。詳細は割愛するが、いずれのコーディング規約も対象とする言語ごとに、その注意すべき点をルールの形で整理しており、また、MISRA-Cのように高信頼性を要求されるソフトウェアを主たる対象にしているもの、対象とする分野や言語等を限定しているものも少なくない。このため、これらのコーディング規約を利用する際にはそれぞれのコーディング規約がどのような特徴を持っているか、どのような開発を対象にしているかを理解して利用することが求められる。

4.3 MISRA-Cとは

近年、MISRA-Cというコーディング規約が話題となっている。

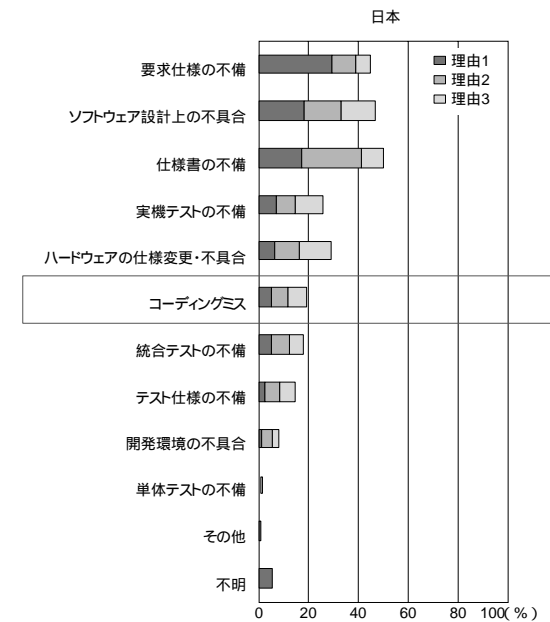


図2 産業実態調査結果1

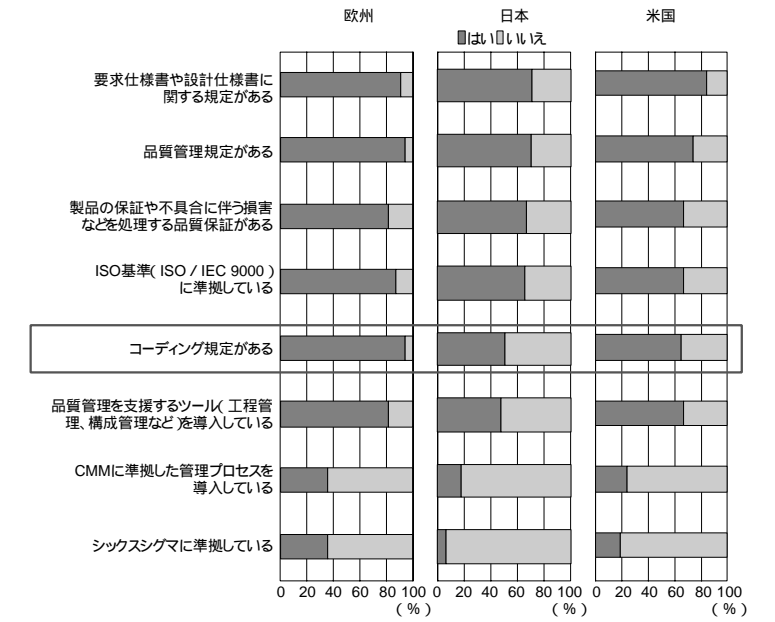


図3 産業実態調査結果2

基準名称	目的	開発年度	特色
GNU Coding Standards	クリーンで矛盾の無いインストールが容易なGNUシステムを作成すること。	2005/01	ドキュメントや、ソフトウェアのインストール、Makefileの規約まで論じられている。
Linux kernel coding style	Linux カーネルのソースコードの好ましい書き方	2004/02	Linux kernelのソースに付属<Linux kernelのソース>/Documentation/Coding Style
Java Code Conventions (Sun Microsystems, Inc.)	Java言語の標準的なコーディングスタイルの取り決め	1999/04	Sun Microsystems, Inc.が作成したコーディングスタイルの取り決め
Programming in C++, Rules and Recommendations (Ellemtel)	C++プログラミングの1つのスタイルを定義する	1992	AT&TのC++言語システムに基づいている。
Recommended C Style and Coding Standards (Indian Hill)	移植性を高め、メンテナンスの手数を減らし、プログラムを分かりやすくする。	1990/06	Cプログラムのために推奨された標準のコーディング。
C STYLE GUIDE (SOFTWARE ENGINEERING LABORATORY SERIES SEL-94-003)	Cプログラムを書くため、ソフトウェア工学研究室(SEL)が推奨するスタイルを記述、ここで、「良いコードとは、組織化した、読みやすい、理解しやすい、維持しやすい、効率的なコードと定義される。	1994/08	ソフトウェア工学原則が論議されて、示され、コード例題が、よい慣例を示すために提供される。
MISRA-C	車載用ソフトウェアを対象としたC言語プログラミングガイドライン。	1998	ソフトウェア開発プロセス全体についての開発ガイドラインがあり、この流れ上での、C言語でプログラムを作成する工程についての開発ガイドライン。127のコーディングルールを規定している。
Safer-C (書籍)	安全関連ワークでのC言語の利用ガイダンス	1995	ソフトウェア品質について述べられている
Effective C++(書籍)等	C++, STL を正しく使う	1995 - 2001	オブジェクト指向デザイン、クラス・デザインにも言及
Coding Style (Embedded@IITD)	Department of Computer Science & Engineering, IITDelhiというグループのコーディング規約	2004	ローカルな規約である。
SPARK 95	完成度の高いアプリケーション・ソフトウェアを書くための高水準プログラミング言語	2005/01	Ada言語のサブセット。言語のみでなくツールセットを含む。
Writing Robust Java Code (The AmbySoft Inc. Coding Standards for Java)	堅実なJavaコードを書く	2000/01	関連書「The Elements of Java Style」 Vermeulen, Ambler, Bumgardner, Metz, Misfeldt, Shur, & Thompson」では、より広い範囲のガイドラインを提示している。
Mozillaコーディングスタイルガイド	Mozilla コードベースで使われる基本スタイルとパターン。	2002/10	Mozilla コードベースで使われる基本スタイルとパターン。
High Integrity C++ Coding Standard Manual (PRQA)	Programming Research Ltdの高品質C++ソフトウェアを開発するためのコーディング標準	2004/05	ルール集。ルールとガイドラインに分類される。
Javaコーディング規約2004 (電通国際情報サービス)	Java言語のコーディング規約を定義する。	2004/09	「超図解 Javaルールブック」は関連図書
Javaコーディング標準 (オブジェクト倶楽部)	Java言語のコーディング規約を定義する。	2002/11	VB.NET, C++, C#に適用されている。

図4 SEC標準調査総合報告書(2004年度版)より

MISRA-Cは、英国の自動車団体(MIRA)が中心となって作成した自動車用ソフトウェア向け、C言語ソースコードの記述ガイドラインである。このガイドラインには、C言語の記述に関して127種類のルールが定められているのが特徴であり、これらはC90を基準としている。MISRA-Cをプログラミング作業の参考として利用することで、理解不能なコードやテスト不可能なコードを書いたりすることを防止したり、テスト時のステートメント・カバレッジを高めるといったことも可能となる。

現在、MISRA-Cは安全が重要視されている自動車、制御、通信、医療、航空宇宙等の分野で組み込みソフトウェアを書くプログラマに広く利用されている。

5. 既存のコーディング規約の課題点

先に欧州ではコーディング規約整備率が90%と紹介した。しかし欧州では、こうした高いコーディング規約整備率にもかかわらず、コーディングミスによる手戻りの発生が90%に至っている。本来、規約が強く守られることによって、品質確保、品質向上が行われると考えられていたが、このデータはコーディング規約だけでは不十分であることを示していると読むことができる。このことは、たとえコーディング規約が存在したとしても、その運用如何によってその効果は大きく変わることを意味しており、この状況を打開するためにはマネジメント面やツール支援等も含めて、総合的に手当てすることが最良の策であると考えられる。以下、コーディング規約に関わる課題点を整理してみたい。

(1) コーディング規約のカバレッジの問題

コーディング規約では規約でいくつぐらいのルールが含まれるか、その粒度も含めたカバレッジが極めて重要になる。即ち、コーディング規約の厳密性を高めようとするとルール数が増え、規定した全てのルールを守ることが難しくなってしまう。一方、ルールのあいまい性を認め緩めのルールを規定すると、ルール数こそ少なくなるものの、逆にルールの抜けや取りこぼし、あるいは具体性(厳密性)が不足して、不具合を招いたり見逃したりする恐れが発生する。このようにコーディング規約では、ルール数やその抽象度合いによって運用方法や得ら

れる効果は大きく変わってしまう。しかし、現状でコーディング規約を策定する場合、どの程度の抽象度、どの程度のルールセット数といった適切な目安が提示されていないため、経験と勘に基づいている場合がほとんどである。

(2) コーディング規約と品質の関係が不明瞭

コーディング規約で規定されたひとつひとつのルールには、それぞれ理由が存在する。そしてその多くは、そのルールを守らないと、何らかの意味でソースコード品質の低下という事象を招くものと考えられる。しかし、通常のコーディング規約ではそのルールひとつひとつの意味を記述しているものは少なく、結果としてコーディング規約の利用者であるプログラマは、

- ・なぜコーディング規約に記載されている項目を守らなければならないのか
- ・なぜコーディング規約が必要なのか

等の背景知識が抜け落ち、結果として個々のルールを守るようとする意識が低下してしまう。そして個々のルールのどの内容がどういった品質につながるかを意識せずにプログラミングを行うこと等も発生し、ソースコードの品質低下を招くことも少なくない。

(3) コーディングに対する意識とスキルの問題

例えば、C言語のポインタ操作やポインタ演算は熟練者にとっては大変に便利な機能である反面、プログラミング初心者にとっては過ちを犯しやすい言語仕様である。このように、一部の言語仕様やプログラミングテクニック等は、プログラマのスキルに応じて使い分ける工夫が必要になる。このため、コーディング規約といっても、対象とする技術者のスキルレベルを考慮してルールセットを設定する必要がある。しかし、現状では、技術者レベルをコーディング規約の中でどのように扱っていくかといった指針や考え方が明確になっていない。

(4) コーディング規約のテラリングの問題

コーディング規約が対象とするソフトウェアは多岐にわたる。例えば、自動車制御等の高信頼性を要求されるソフトウェアのソースコードで注意すべき点と、ユーザ操作を中心とするUI部分のような操作性や使用性が重視されるソフトウェアのソースコードで注意すべき点は異なっている。このように異なる側面を持つソフトウェアに対し、一律のルールを適用することは現実的ではない。

このため、コーディング規約を準備する場合には、対象とするソフトウェアの特性、あるいは適用部門の特性等を把握し、適切なルールを選択した上で必要に応じてコーディングルールを調整するテラリング作業が必須である。しかし、これまでのコーディング規約では利用者のコンテキストや対象ソフトウェアの特性に合わせたルールのテラリング方法まで踏み込んでいるものは少なく、運用でカバーしている。

6. ソースコード品質向上のためのコーディング作法

ソースコードの品質を向上させるための道具としての現状のコーディング規約には、上記のような課題点があり、結果として必ずしも十分に機能していないものも少なくない。SECではこのような課題認識のもと、実際の開発現場で容易に導入でき役に立つコーディング規約をどのようにしたら整備できるかについて議論している。これまでの議論の中から、

既にある様々なコーディング規約やルールを調査し、それらを利用する側で取捨選択可能な形に整理する。

上記整理の過程で、個々のルールの意味や意義の理解を容易にするために、前述の品質特性モデルの枠組みを利用した階層構造で整理する。

実際の開発現場のコンテキストや対象ソフトウェア特性に合致するようにルールを選択したり調整(チューニング)可能なように、規約のテラリング方法を明示する。

といった特徴を持つ「コーディング作法ガイド」の策定を進めている。この「コーディング作法ガイド」は主にコーディング規約を作成する際に、どのような考え方でルールセットを決めていくかを明示し、またその際に参考とすべき参照ルールセットのデータベースを併せ持っている。このため、これから新たにコーディング規約を定めたり、あるいは、既にある社内コーディング規定を改定する際の参考として利用していただくことを想定している。また、参照ルールセット・データベースは、MISRA-Cを始めとする多数のコーディング規約をベースとしているため、新人へのプログラミング技術教育の参考としても利用できると考えている。「コーディング作法

ガイド」の詳細解説については別の機会に譲るが、現在、この「コーディング作法ガイド」の一部をドラフト版(V0.8)としてSECホームページ上で公開し、パブリックコメントを募集している。本解説をご覧になり興味をお持ちになった方は、是非、ご覧いただきご意見をいただければ幸いです。なお、「コーディング作法ガイド(V0.8)」は皆様からのコメントを参考に、今年度、より洗練し、広く利用していただける形で公開を予定している。

参考文献

2004年組込みソフトウェア開発実態調査報告, 経済産業省
 MISRA-C研究会: 組込み開発者におけるMISRA-C組込みプログラミングの高信頼化ガイド, 日本規格協会, 2004年
 プログラミング作法, ASCII, 2001年
 C言語診断室, 技術評論社, 2003年
 叫ぶ! Cプログラマ, ソーテック社, 2004年
 プログラミング言語C, 共立出版, 2004年
 C/C++プログラミング, プレンティスホール出版, 1988年
 CプログラミングFAQ Cプログラミングのよく尋ねられる質問(新紀元社情報工学シリーズ), 新紀元社, 2004年
 Cプログラミングの落とし穴(新紀元社情報工学シリーズ), 新紀元社, 2004年
 エキスパートCプログラミング 知られざるCの深層(Ascii books), 新紀元社, 1996年
 Cスタイル標準とガイドライン, 海文堂出版, 1993年
 ライティングソリッドコード バグのないプログラミングを目指して(マイクロソフトプレスシリーズ), ASCII, 1995年
 現実的なC++プログラミング C++ Strategies and Tactics (ADDISON WESLEY プロフェッショナルコンピューティングシリーズ), ソフトバンク, 1994年
 注解C++リファレンスマニュアル, シイエム・シイ, 2001年
 ANSI C/C++辞典, 共立出版, 1996年
 C/C++によるプログラミングスタイルブック, ソフトバンクパブリッシング, 2000年
 C/C++ミスプログラミング ケーススタディ (Front Programmer Series), 秀和システム, 2003年
 プログラミングの基礎(情報科学こんせぶつ), 朝倉書店, 1997年
 Cプログラミング書法 移植性の高いプログラムを書くために, OQ出版, 1996年
 新版 C言語プログラミングのエッセンス, ソフトバンクパブリッシング, 1996年
 文芸的プログラミング (ASCII SOFTWARE SCIENCE Programming Paradigm), ASCII, 1994年
 コードコンプリート 完全なプログラミングを目指して (Microsoft PRESS), ASCII, 1994年
 超図解 Javaルールブック, エクスメディア, 2004年
 The Elements of Java Style (Advances in Object Technology, 26), Cambridge University Press, 2000年
 Effective Java プログラミング言語ガイド, ピアソン・エデュケーション, 2001年

* なお、本稿で使用されている標準調査は仮収集結果なので、公開版とは違う可能性があります。

PMBOKの概要

組込みソフトウェア開発力強化推進委員会 プロジェクトマネジメント技術部会 委員 / 日本電気株式会社 エラーニング事業部 プロセスコンサルティンググループ 兼 組込みソリューション事業推進本部 マネージャ 工学修士 (情報工学)

井沢 澄雄

ITプロジェクトにおいて、米国PMI (Project Management Institute) が発行しているプロジェクトマネジメント知識体系ガイド (A Guide to the Project Management Body of Knowledge、略称: PMBOKガイド) の利用が始まっている。エンタプライズ系ソフトウェア開発での利用が先行しているが、組込みシステム開発プロジェクトにおいても、組込み特有のマネジメントスタイルを取り込んだ形での利用が期待されている。

本稿では、2004年に発行されたPMBOKガイド第3版に含まれる44個のプロジェクトマネジメントプロセスと、プロジェクトマネジャーの共通用語とすべき重要単語について概説を行う。

1. はじめに

米国の非営利団体であるPMI (Project Management Institute) が発行しているプロジェクトマネジメント知識体系ガイド (略称: PMBOKガイド) [1]が、IT企業のプロジェクトマネージャに普及してきている。同ガイドは、1996年、2000年にそれぞれ、第1版、第2版が発行され、2004年に発行された第3版が最新のものとなっている。

次節以降では、エンタプライズ系システム開発プロジェクト (以下、エンタプライズ系) と組込みシステム開発プロジェクト (以下、組込み系) における留意点も盛り込み、PMBOKガイドに記載されている主なプロジェクトマネジメントプロセスの解説を行う。

2. プロジェクトとプロジェクトマネジメント

PMBOKガイドによるプロジェクトの定義は次のとおりである。

プロジェクトとは、独自のプロダクト、サービス、所産 (result) を創造するために実施される有期性の業務である。

定義にある「独自性」と「有期性」が、これまでとは違う何らかの活動を新規に行わなければならないという、プロジェクトの宿命を明確に表現している。そして、それゆえに、プロジェクトというものは何らかの不確実性

を内在しており、適切なプロジェクトマネジメントの実行が不可欠となっている。

なお、「創造」物とは、エンタプライズ系ではビジネスアプリケーションが、組込み系では製品へ組み込まれるシステム (ハードウェア+ソフトウェア) がそれにあたる。

次に、PMBOKガイドによるプロジェクトマネジメントの定義を示す。

プロジェクトマネジメントは、プロジェクトの要求事項を満足させるために、知識、スキル、ツールと技法をプロジェクト活動へ適用することである。

プロジェクトマネージャは、プロジェクトの目標を達成する責任を負っており、プロジェクトに対する種々の要求事項を満たすため、PMBOKガイド等によって紹介されるプロジェクトマネジメント技術をタイミングよく適用していくことが求められる。

なお、組込み系では、ハードウェア開発と組込みソフトウェア開発の2つのプロジェクトの整合をとったマネジメント活動を必要とするが、PMBOKガイドでは複数プロジェクトの調和をとるマネジメントを「プログラムマネジメント」と呼んでいる。また、プログラムマネジメントの上位概念として、「ポートフォリオ・マネジメント」を置き、戦略的なビジネス目標を達成するために、各プロジェクトやプログラムをグループ化する考え方も示している。たとえば、組込み系において、複数の組込みシステム製品を開発するプロジェクト毎に、リスクと期待利益の対比を行い、資源や資金の割り当ての優先順位を決定すること等がこれに当たる。

3. プロジェクトマネジメントプロセス

PMBOKガイドでは、プロジェクトマネジメントは一連のマネジメントプロセス (第3版では44個) を通して実行されていくこととされており、それらは、以下に示す5つのプロセス群のいずれかに含まれる。なお、これらのプロセス群は、プロジェクトの開始から終了までの全期間を通じて、オーバラップして実行されることに注意が必要である。

立上げプロセス群 (2プロセス)

新しいフェーズやプロジェクトの開始 (もしくは再開) を公式に認可するためのプロセスのグループ。

計画プロセス群 (21プロセス)

プロジェクトの目標を達成するための一連の活動を計画し、プロジェクトマネジメント計画書としてまとめるためのプロセスのグループ。

実行プロセス群 (7プロセス)

人と資源と情報を調整、統合し、プロジェクトマネジメント計画書に盛り込まれた活動を実行するためのプロセスのグループ。

監視コントロールプロセス群 (12プロセス)

プロジェクトの実行を監視して計画からの差異を識別し、プロジェクト目標を達成するための種々の是正処置を実行するプロセスのグループ。

	1 立上げプロセス群	2 計画プロセス群	3 実行プロセス群	4 監視コントロールプロセス群	5 終結プロセス群
1 統合マネジメント	☆	☆	☆	☆	☆
2 スコープマネジメント		☆		☆	
3 タイムマネジメント		☆		☆	
4 コストマネジメント		☆		☆	
5 品質マネジメント		☆	☆	☆	
6 人的資源マネジメント		☆	☆	☆	
7 コミュニケーションマネジメント		☆	☆	☆	
8 リスクマネジメント		☆		☆	
9 調達マネジメント		☆	☆	☆	☆

図1 マネジメントプロセス群と知識エリア

終結プロセス群 (2プロセス)

フェーズやプロジェクトのすべての活動を公式に終了させ、プロジェクトの最終成果物を引き渡すためのプロセスのグループ。

4. プロジェクトマネジメント知識エリア

44個のプロジェクトマネジメントプロセスはまた、それが関連する知識エリアによって、9つのいずれかのエリアに分類される。図1は、マネジメントプロセス群と知識エリアの組合せを示しており、44個のプロセスのいずれかが含まれる箇所に印を記している。

また、各マネジメントプロセスには図2に示すような、標準的な順序関係がある。図中の「4.1」、「4.2」といった

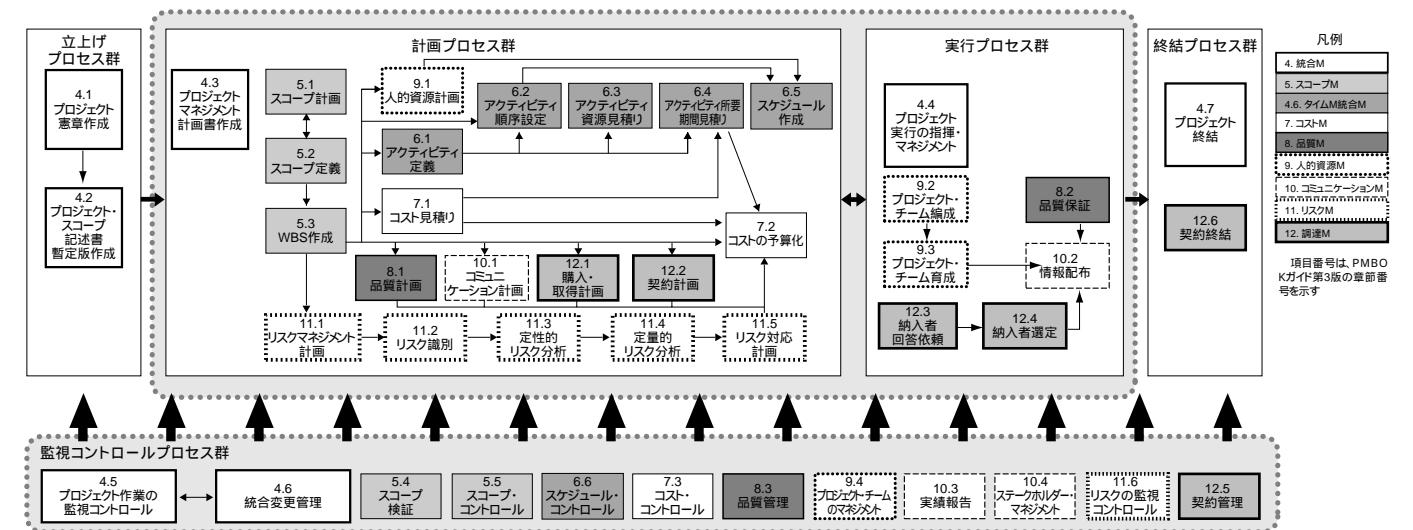


図2 プロジェクトマネジメントプロセスの関連

Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide) Third Edition Official Japanese Translation, Project Management Institute, Inc., 2004. Copyright and all rights reserved. Material from this publication has been reproduced with the permission of PMI ©.

項目番号はPMBOKガイド（第3版）の章節番号を表す。

個々のマネジメントプロセスの概要を以下で説明するが、一部、エンタプライズ系、組込み系についての補足を行っている。なお、プロセス名の末尾に付記してある【立上げ】【計画】【実行】【監視】【終結】は、当該プロセスが含まれるプロセス群を示す。

4.1 プロジェクト統合マネジメント

プロジェクト憲章作成 - 【立上げ】

プロジェクトを公式に認可するためのプロジェクト憲章を作成する。プロジェクト憲章の発行により、プロジェクトマネージャに組織の資源を使用する権限が与えられる。受託開発を行うエンタプライズ系では顧客との契約書が、組込み系では企画部門等から提供される企画書等が、作成のための主な入力情報となる。

プロジェクト・スコープ記述書暫定版作成 - 【立上げ】

プロジェクトを定義するためのプロジェクト・スコープ記述書を作成する。プロジェクトによって創出される成果物の目的、要求事項、要素成果物（次節で説明）、プロジェクト組織、識別されたリスク等を記載する。

プロジェクトマネジメント計画書作成 - 【計画】

個別に作成する複数の計画書を統合して、プロジェクトマネジメント計画書を作成する。

プロジェクト実行の指揮・マネジメント - 【実行】

プロジェクトマネジメント計画書で規定された活動の実行を指揮し、技術間や組織間のインタフェースをマネジメントする。組込み系では要素技術や関連組織が比較的多いため、きめの細かいマネジメントが必要となる。

プロジェクト作業の監視コントロール - 【監視】

プロジェクト活動を監視し、状況に合わせて予防措置や是正措置を行い、プロジェクト目標の達成を目指す。

統合変更管理 - 【監視】

プロジェクトの実行に伴って発生する様々な変更要求を統合的にマネジメントする。ある領域（スコープ等）の変更要求が他の領域（コスト、スケジュール等）の変更を発生させるため、変更の全体調整が必要となる。

プロジェクト終結 - 【終結】

要素成果物の公式受入を行い、プロジェクトまたはフェーズを公式に終了する。

4.2 プロジェクト・スコープ・マネジメント

スコープ計画 - 【計画】

プロジェクト・スコープ（プロジェクト目標を達成するために実行する必要のある作業群）の定義方法、検証方法、マネジメント方法等を記載したプロジェクト・スコープ・マネジメント計画書を作成する。

スコープ定義 - 【計画】

プロジェクト立上げ時に作成した暫定版のプロジェクト・スコープ記述書をベースに、その後明らかになった情報を加味して、より詳細なプロジェクト・スコープ記述書を定義する。

WBS作成 - 【計画】

要素成果物を創出するためのプロジェクト作業を特定するため、要素成果物をマネジメントしやすい粒度まで階層的に要素分解（細分化）する。WBSについては、次節で説明を行う。

スコープ検証 - 【監視】

プロジェクト・スコープの完了を検証するため、ステークホルダー（利害関係者）から、要素成果物の公式な承認を受ける

スコープ・コントロール - 【監視】

プロジェクト・スコープに対する変更要求をマネジメントし、影響を適切にコントロールする。

エンタプライズ系では顧客との仕様変更や作業分担変更等の手順を事前に合意しておくことが、適切なスコープ・コントロールには不可欠である。

4.3 プロジェクト・タイム・マネジメント

アクティビティ定義 - 【計画】

プロジェクトにおいて実行されるべきすべての作業を特定する。プロジェクトメンバが理解できる程度の詳細さで記述する必要がある。ローリングウェーブ計画法（次節で説明）を用いて段階的に詳細化する。

アクティビティ順序設定 - 【計画】

上記で定義されたアクティビティに順序関係を設定する。順序関係の記述には、プレシデンス・ダイアグラム法（PDM）等の表記法が用いられる。

アクティビティ資源見積り - 【計画】

各アクティビティを実行するために必要となる資源（人、機器、資材）の種類（スキル等）と量（人数等）を

見積る。また、当該資源の割当可能期間も明確にする。

アクティビティ所要期間見積り - 【計画】

必要な資源の種類と数量、資源の割当可能期間等の情報を基に、アクティビティ毎の所要期間を見積る。

スケジュール作成 - 【計画】

各アクティビティの開始・終了予定日を決定する。クリティカルパスを識別し、クラッシングやファストトラッキング等のスケジュール短縮技法（次節で説明）を適用する。

スケジュール・コントロール - 【監視】

スケジュール・ベースラインに対する変更要求をマネジメントし、影響を適切にコントロールする。

4.4 プロジェクト・コスト・マネジメント

コスト見積り - 【計画】

各アクティビティの実行に必要な資源の概算コストを見積る。見積りにあたっては、リスクを考慮して、変動の要因を検討する。

コストの予算化 - 【計画】

各アクティビティの見積りコストを取りまとめ、コスト・ベースラインを設定する。コスト・ベースラインは、時系列の累積コスト金額として表される。

コスト・コントロール - 【監視】

コスト・ベースラインに対する変更要求をマネジメントし、影響を適切にコントロールする。

4.5 プロジェクト品質マネジメント

品質計画 - 【計画】

プロジェクトに関連する品質規格を特定し、それを満たすための計画を策定する。

品質保証 - 【実行】

プロジェクトに対する要求事項を満たすために必要なプロセスを確実に実行するため、計画的で体系的な品質保証活動を行う。

品質管理 - 【監視】

プロジェクトの実行結果が適切な品質規格に適合していることを確認する。

4.6 プロジェクト人的資源マネジメント

人的資源計画 - 【計画】

プロジェクトに必要な人的資源を検討し、各メンバに

役割と責任を割り当て、報告関係を明確にする。

プロジェクト・チーム編成 - 【実行】

人的資源計画の計画に従い、プロジェクトに必要なメンバを確保する。

プロジェクト・チーム育成 - 【実行】

メンバのコンピテンシー強化とメンバ間の交流を促進し、プロジェクトのパフォーマンス向上を図る。

プロジェクト・チームのマネジメント - 【監視】

メンバのパフォーマンスを追跡し、パフォーマンス向上のための課題解決を促進する。

4.7 プロジェクト・コミュニケーション・マネジメント

コミュニケーション計画 - 【計画】

各ステークホルダーが、いつ、どのような情報を必要とし、その情報を誰がどのように提供するか等を検討し、文書化する。

情報配布 - 【実行】

プロジェクト実行中に、ステークホルダーが必要とする様々な情報を、適切なタイミングで提供（計画的提供、随時の要求に対する提供）する。

実績報告 - 【監視】

すべてのベースライン・データを収集し、ステークホルダーに対してその時点でのプロジェクトパフォーマンス情報（実績測定ベースラインとの比較分析結果を含む）を提供する。アーンド・バリュー分析（次節で説明）を活用するケースも増加の傾向にある。

ステークホルダー・マネジメント - 【監視】

ステークホルダーのニーズを満たし、ステークホルダー間の課題を解決するために、ステークホルダー間のコミュニケーションをマネジメントする。

4.8 プロジェクト・リスク・マネジメント

リスク・マネジメント計画 - 【計画】

どのようにリスク・マネジメント活動を行うかを計画する。

リスク識別 - 【計画】

プロジェクトに影響を与えるリスクを識別する。エンタプライズ系、組込み系でそれぞれ留意すべきリスク領域に共通性（エンタプライズ系：顧客関連リスク、組込み系：ハードウェアとの並行開発関連リスク/マトリクス組織関連リスク等）もみられ、リスクチェックリスト

も効果的である。

定性的リスク分析 - [計画]

上記で識別したリスクに発生確率と発生時の影響を割り当て、対応の優先付けを行う。

定量的リスク分析 - [計画]

リスクの影響を、シミュレーション等を用いてより詳細に分析する。

リスク対応計画 - [計画]

識別されたリスクに対する対応計画を作成する。対応戦略として、マイナスのリスクに対する回避 / 転嫁 / 軽減と、プラスのリスクに対する活用 / 共有 / 強化、プラス・マイナスどちらのリスクでも利用される受容がある。

リスクの監視コントロール - [監視]

プロジェクト実行中の新たなリスクの識別、識別済みリスクの追跡、残存リスクの再検討、トリガー条件の監視等を行い、プロジェクトの目標に与えるリスクの影響の低減を図る。

4.9 プロジェクト調達マネジメント

購入・取得計画 - [計画]

外部からの調達の必要性を検討し、いつ、何を、どのように、どの程度、取得すべきかを計画する。

契約計画 - [計画]

外注候補から入札書やプロポーザルを獲得するために必要な文書（提案依頼書等）を作成する。

納入者回答依頼 - [実行]

外注候補（複数）に入札依頼やプロポーザル提出依頼

を行い、見積書や提案書を受領する。

納入者選定 - [実行]

受領した入札書やプロポーザルを検討し、納入者を決定する。

契約管理 - [監視]

購入者（発注側）と納入者（受注側）の双方が契約上の義務を果たし、法的な権利が保護されていることをマネジメントする。

契約終結 - [終結]

納入者のすべての作業と要素成果物が受入可能であることを検証し、契約をクローズする。

表1にエンタプライズ系、組込み系のそれぞれにおける知識エリア毎のマネジメント課題例を示す。知識エリア全般に課題が存在していることがわかる。

5. 重要用語

本節では、プロジェクトマネジメントプロセスの理解を促進する重要用語の説明を行う。

5.1 要素成果物 (Deliverable)

フェーズやプロジェクトを完了させるために創出されるもので、固有で検証可能なもの。顧客やスポンサーに承認をもらうためのプロジェクト外部への納品物という狭義の意味で用いられることも多い。

表1 エンタプライズ系、組込み系におけるマネジメント課題

	エンタプライズ系でのマネジメント課題	組込み系でのマネジメント課題
1 統合マネジメント	・顧客との契約書への詳細合意事項の盛り込み ・顧客交渉窓口の明確化	・製品開発全体はプログラムマネジメント、ポートフォリオマネジメント ・プロジェクトのGO / NOGO的確な判断
2 スコープマネジメント	・顧客との仕様確定や仕様変更に伴うコスト増調整等の手順の合意 ・エンドユーザの早期巻き込み ・顧客との実現性のある作業分担(顧客責任者からの承認印)	・システムレベル設計による、ハードウェアとソフトウェアの分担調整 ・スコープ変更結果情報の即時のステークホルダー通知 ・マネジメント活動の事前計画
3 タイムマネジメント	・顧客都合によるスケジュール変更手順の合意	・対象機器との並行開発やコ・デザイン、クロス開発を考慮したスケジュール設定
4 コストマネジメント	・見積り精度向上 ・利益を意図したコスト管理	・組織としてのコストマネジメント ・予算把握の徹底
5 品質マネジメント	・上級フェーズからの品質作り込み ・顧客要求仕様品質の向上	・絶対品質確保のための品質計画、高信頼性の確保、フェールセーフ ・リアルタイム性の検証、リモートデバッグ、ICEの活用
6 人的資源マネジメント	・業務スキル保持者の確保 ・必要タイミングでの必要リソース量の確保	・機能部門との折衝による開発メンバの獲得 ・電気・電子・力学の基礎知識や使用プロセッサ / OS、UML、JAVA等のスキル習得
7 コミュニケーションマネジメント	・顧客(契約担当、システム担当、エンドユーザ)とのコミュニケーション ・営業担当との連携 ・指示系統の明確化 ・定量的な進捗把握	・機器製作者との正確なコミュニケーション ・プロジェクト上部へのタイムリーなエスカレーション ・プロジェクト状況を関係者(ハード技術者、ソフト技術者、協力会社等)からの確に聞き出すコミュニケーションスキル
8 リスクマネジメント	・顧客関連リスク、新技術リスク、複数モデルウェアの組み合わせリスク ・営業担当との共通リスク認識 ・コンタインジェンシー予備(顧客要因リスクの顧客転嫁)	・ハードウェア開発側とのインタフェースに関するリスク、外注に関するリスク ・プロジェクト横断的なリスク対応策(他プロジェクトへの動的メンバ再割当等) ・製造物責任(PL)法を意図したリスクマネジメント
9 調達マネジメント	・業務知識保持者の確保、最大価値を発揮してもらえる環境作り ・複数プロジェクトにわたる、優秀な外注メンバの継続的確保	・ツール共有(構成管理、プロジェクト管理、品質管理等) ・契約形態、契約書の文言によりリスク軽減

5.2 WBS (Work Breakdown Structure)

要素成果物を段階的に要素分解したものであり、プロジェクトの目標を達成するために実行しなければならない作業の集まりである。表記のレベルが1段下がるごとに、作業をより詳細に定義する。

5.3 クリティカルパス

プロジェクトのアクティビティに順序付けを行って作られたスケジュール・ネットワークにおける、プロジェクトの開始から終了までの最短経路のこと。クリティカルパス上のアクティビティが少しでも遅れると、プロジェクト全体のスケジュールが遅れてしまうため、常に注意を払っておくべき経路である。

クリティカルパスの算出では資源に関する制約を考慮しないが、並行で実行されるアクティビティに同一メンバが割り当てられることを避けるようにスケジュール修正を行うクリティカル・チェーン法が第3版から記載されている。

5.4 クラッシング、ファストトラック

スケジュールの短縮技法の一種である。プロジェクト全体の期間を短縮するために、クラッシングでは、クリティカルパス上のアクティビティに要員を追加し、ファストトラックでは、クリティカルパス上の作業を細分化して並行実行を行う。なお、いずれの方法にも、コスト増や手戻りのリスクという負の側面があるため、実行した場合の効果をしっかり分析しておく必要がある。

5.5 ローリングウェーブ計画法

計画を段階的に詳細化していく計画手法。遠い将来の作業はWBSの上位レベルの計画にとどめ、近い将来の作業はWBSの下位レベルまで詳細に計画する。段階的詳細化を行う際には、スコープの無秩序の拡大(スコープ・クリープ)を招かないようにコントロールを行う必要がある。

5.6 アード・バリュ法

プロジェクトの実績を、ブランド・バリュー(PV)、アード・バリュー(EV)、実コスト(AC)という3つの値を用いて分析、評価する技法。スケジュール、コス

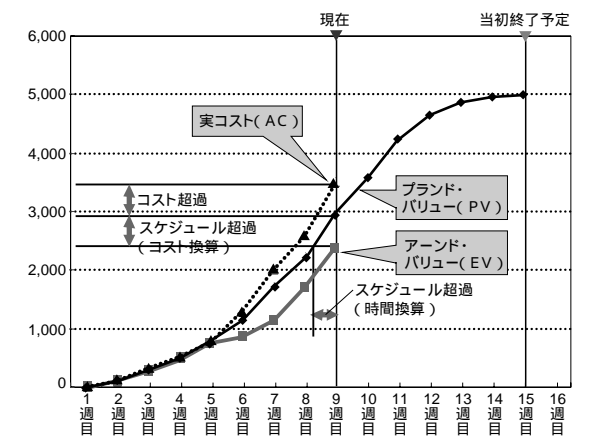


図3 アードバリュ分析の例

トの両面から、現時点での進捗評価とプロジェクト完了時の予測を行う。図3にアード・バリュ法によるコストとスケジュールの評価例を示す。

6. 組込みソフトウェア開発とPMBOK

今、組込みソフトウェア開発の現場では、機能規模の急激な増大と、それに伴う開発組織の肥大化に直面しており、プロジェクトマネジメントの必要性が急速に高まっている。そのような状況の下、PMBOKガイドによって共通用語を理解し、プロジェクトマネジメント活動を実践することは非常に有益である。

組込みソフトウェア開発力強化推進委員会の下部組織であるプロジェクトマネジメント技術部会では、PMBOKガイド等の各種標準を参照しつつ、要素間のインタフェースを「すり合わせ」ながら開発してゆく日本の組込みソフトウェア開発に適したプロジェクトマネジメント手法の整備等を目標とした活動を行っている。

今後、エンタプライズ系、組込み系の両分野のプロジェクトマネージャが共通用語で情報交換を行い、それぞれが得意とするマネジメント手法を融合した日本流の新しいプロジェクトマネジメント手法が生み出されていくことを期待したい。

参考文献

[1] Project Management Institute, Inc.: プロジェクトマネジメント知識体系ガイド 第3版, 2004
 [2] 組込みソフトウェア開発におけるプロジェクトマネジメントの動機(ドラフト版), 独立行政法人情報処理推進機構, 2004

要求工学への期待

エンタプライズ系ソフトウェア開発力強化推進タスクフォース 要求工学研究会 委員
株式会社野村総合研究所 流通システムプロジェクト部 主任
荒生 知之

ソフトウェア開発を取り巻くビジネス環境や技術環境が大きく変化中、ソフトウェアに対する戦略的価値がより求められるようになってきている。それと同時に、ソフトウェアの要求を仕様化する工程の困難性も増加し、数多くのプロジェクトが要求に起因した失敗を繰り返している。そのような産業界の課題を扱う学問分野として、要求工学は今後さらにその重要性を増していくだろう。しかし要求工学が取り扱う分野は特に人間系の作業が多い部分であり、技術の適用が難しい分野でもある。本稿では、要求工学の現場適用の事例を踏まえ、要求工学を業界にインパクトをあたえる実践的な分野に発展させていくための方向性について述べる。

1. 要求工学を取り巻く環境の変化

ソフトウェア開発における要求定義の重要性が認識されるにいたって久しいが、昨今のソフトウェアを取り巻く環境の変化に伴い、その重要性はさらに高まりつつある。

ビジネス面においては、各企業を取り巻く事業環境・経営環境の変化がより激しくなり、経営に戦略性がより求められる中で、ソフトウェアに対しては変化に対する素早い適応力が求められるようになってきている。また、技術面においては、ハードウェア性能・ソフトウェア技術が大幅な発展を遂げ、複雑かつ大規模なシステムの実現が技術的に可能になっている中で、ソフトウェアの役割も従来の業務効率化や情報活用支援から、事業・サービス創造や業務

プロセス標準化支援など、より戦略的かつ未知・未経験業務のサポートに変化しつつある(図1)。

このようにソフトウェアに求められる役割が変化中、ソフトウェア要求を仕様化するプロセスも、単に既存業務を支援するためにソフトウェア機能の要求を定義するというプロセスから、経営・エンドユーザからのニーズに応じて新たな業務を設計し、それをソフトウェアの要求仕様に変換していくプロセスへとその質を変化させている。

だが同時に、ソフトウェアの要求仕様の不備を原因とするプロジェクトの失敗は減る傾向を見せない。JUAS(社団法人 日本情報システム・ユーザー協会)の2003年度ユーザ企業情報化動向調査でも、発注者側から見たシステム開発上のトップ反省要因として実に6割以上もの要因が要求の仕様化のプロセスにあるとの結果が出てい

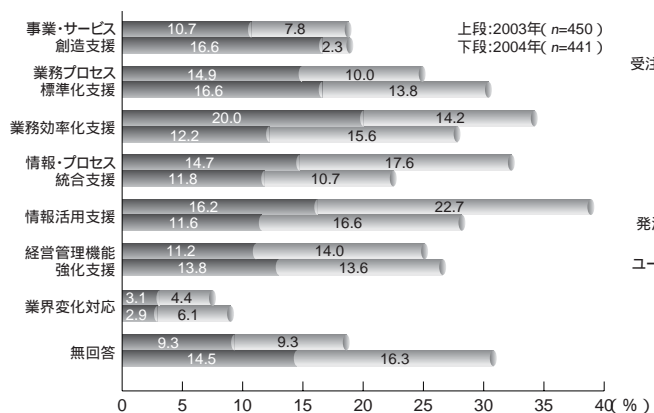


図1 重視するITの活用目的

出典：野村総合研究所「ユーザ企業のIT運営実態調査」2004年11月

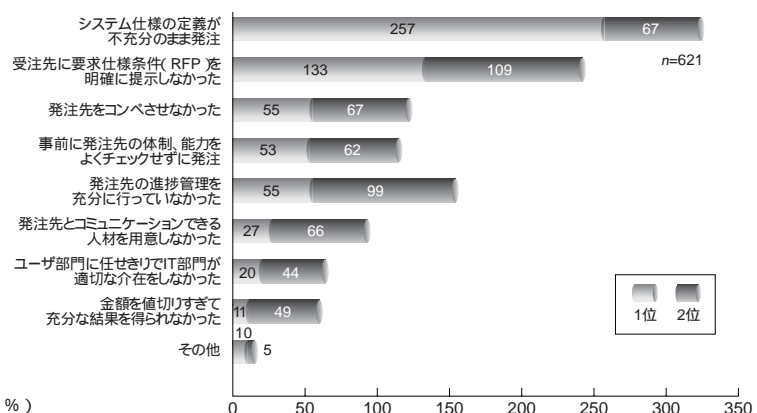


図2 発注者側から見たプロジェクトの反省要因

出典：JUAS ユーザ企業情報化動向調査2003

る(図2)。現場でプロジェクトを実施している立場からも、業務の視点から要求を十分に獲得、検証、管理できていないことが、ますますソフトウェア開発の品質・生産性の改善を阻害させてきていると実感している。

2004年11月16日に開催されたSECの設立シンポジウムにおいても、『ITシステムを求める人』と『ものづくりの人』とのギャップ」が広がる中で、「それを埋めるための開発に入るまでのプロセスの明確化/取り込み」がSECに対する主要な期待として述べられる等[1]、ソフトウェア要求の仕様化の工程に対する改善要望が多くあげられていた。

このように要求を仕様化するプロセスは、今日的にその重要性や範囲を増大させていながら、同時にプロジェクトの多くの失敗要因になっており、まさに優先的に取り組むべき分野といえる。要求を仕様化するプロセスを工学的視点で研究する学問領域として要求工学(Requirements Engineering) (囲み記事参照)が注目されつつあるのは、時代の必然であるといえよう。

本稿では、弊社流通システムの設計・開発を主業務とする事業部門から立ち上げた要求工学適用研究プロジェクトの事例を通じて、ソフトウェア開発現場の視点での要求を仕様化するプロセス(以下要求工学プロセス)における課題とその原因、また、それに対して要求工学が果たすべき役割と要求工学に対する期待を述べる。

COLUMN

要求工学と要求工学プロセス

要求工学とは、「ソフトウェア開発における要求を仕様化するプロセスを工学的に定式化する技術 [2]」であり、利用者の要求の獲得、定義、検証、管理等のプロセス(図3)がその研究対象となる(本稿では、要求工学の研究対象となる上記ソフトウェアのプロセスを総称して要求工学プロセスと定義して使用している)。

ソフトウェア開発における要求定義の重要性は、20世紀の半ばから認識され、ソフトウェア工学の研究対象となってきたが、要求工学という言葉が登場したのは1993年の要求工学国際会議(<http://www.requirements-engineering.org/>)と比較的新しい[3]。本稿では実践的側面について記述しているため、要求工学の学問的側面については、参考文献の[2],[3]などを参照されたい。

2. 改善を妨げているものは何か?

要求工学プロセスの重要性はソフトウェア構築に携わっているほとんどの者が認識していることであろう。また、同プロセスに対しては、これまでさまざまな技術的工夫が提案、適用されてきた。では何故、多くのプロジェクトにおいて、要求工学プロセスを起因とする失敗が未だに繰返されているのだろうか?

野村総合研究所の事例を見てみよう。当社においても、利用する側の視点と作る側の視点のギャップに対する課題意識を背景に、社内標準プロセスにおいて業務設計という独立した活動を定義している。即ち、システム設計を実施するうえでは、まず、要求定義工程の中で業務設計を実施し、業務レベルでユーザの要求を定義する。次に、概要設計の中で、ユーザの要求を実現するための画面や帳票などの機能の構成・概要と方式を設計する。さらに、基本設計では画面設計、内部処理設計など機能単位の処理内容をモジュール単位に落としレベルまで詳細化する手順を標準としている。しかしながら、現場では、業務設計レベルにおいて業務要求は曖昧に定義され、十分時間をかけて実施されていないことが少なくない。

では、ユーザからの業務レベルの要求は実際にはどの工程で抽出され、検証されているのであろうか? 図4

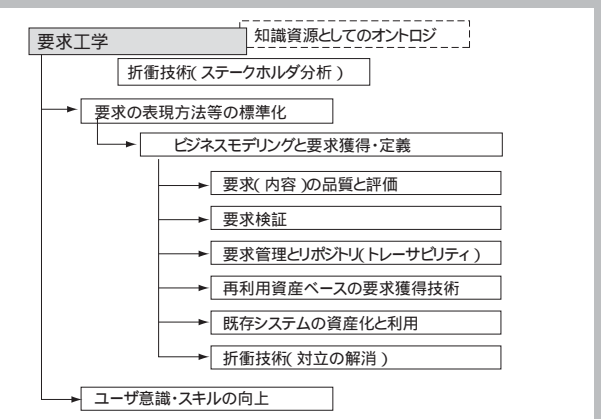


図3 要求工学を構成する技術

は弊社のあるプロジェクトの顧客との打ち合わせ議事録の量（円の大きさ）の推移を工程別に時系列に表したものである。議事録を顧客からの要望の記録であると考えれば、業務レベルの要求も含めて要求の獲得が、基本設計フェーズでもっとも実施されていることが想定される。

実際に現場のプロセスの実態をみると、所謂要求定義フェーズでは、非常に大まかな業務要求のみを定義し、詳細な業務レベルの要求獲得は、設計フェーズでシステム機能を定義する過程で実施されている場合が一般的である。即ち、『ITシステムを求める人』と『ものづくりの人』との間のギャップは、機能の設計のフェーズにおいて、設計者の業務知識・ノウハウに依存しながら埋められている。また、そのギャップを埋めている過程において、『ITシステムを求める人』の要求、すなわち業務レベルの要求は、一部議事録などには残るもの、明確な形で記述されず、形式知化されていないため、資産として共有されにくく、ノウハウとしても引き継がれない場合が多い。いくつかの業務レベルの要求は機能レベルの設計の中に前提として記述されても、機能レベルの単位で記述されているため業務レベルの要求が設計として実現されているかを検証することが難しくなる。

仮に、業務レベルの要求構造とシステムの機能構造が一对一の対になっていれば、両者間にギャップは生じない。しかし、業務要求の構造は業務の文脈に沿って出されるのに対して、システム機能は業務を超えた論理的構造となるため、ユーザ視点からの業務レベルの要求とシステム機能レベルの要求との間には、複雑な直交性が発生してしまう。即ち、1つの業務の単位に対する要求は、必ずしも、1つのシステムに対する要求ではなく、複数のシステムに反映されることがある。逆に、1つのシステムに対する仕様は、複数の業務に対する要求が反映されている。結果として、ある機能のソフトウェアの仕様

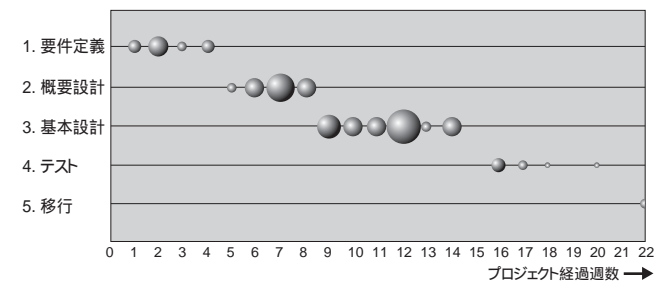


図4 工程別議事録の作成量の推移

をみただけでは、その仕様がどのような要求を元に決定されたものかは判別できない。

業務レベルの要求視点ではなく、機能レベルの視点に基づいて業務要求の獲得が実施されると、業務レベルと機能レベルの間のギャップを埋める作業は属人的に実施されることになるため、業務視点での表現・検証が困難となり、結果として、そのギャップに起因するプロジェクトの失敗は繰り返されることになる。機能レベルの要求が直接出される業務効率化を目的としたプロジェクトであればまだしも、最近増えつつある新業務を前提としたプロジェクトにおいては、システムが具現化してはじめて、業務レベルでの要求が検証可能となり、後工程で顕在化したギャップが、スケジュールの逼迫、トラブルの発生、"使われないシステム"の発生を引き起こす。

上記のように業務レベルの視点を失って設計を進めてしまう根本的な原因には、主に下記のようなものがあるだろう。

- ・業務レベルの要求定義プロセスは規定されていても、それを機能レベルの設計に落とし込む、または相互の関連性を管理する技術・プロセスがない。
- ・ソフトウェア構築においては、ユーザも含め現場は後工程の作業の中心である"ものづくり"の視点でシステムを捉えがちである。
- ・業務が決まらない中で、スケジュール上の制約から設計を先行せざるを得ない。

上記に挙げた原因は、プロジェクトの種類・性質によって様々であるが、このように実際の現場のプロセスをミクロ的視点で分析すると、要求に起因する失敗が繰り返される根本的なボトルネックは、『ITシステムを求める人』と『ものづくりの人』との間のギャップを埋めるためのプロセスが様々な原因を背景として属人的に実施されているため、業務レベルの要求の不整合性、抜け・漏れが後工程まで顕在化しないことにあるケースが目立つ。このような課題は、他の企業でも共有されるものであろう。

3. 要求工学の適用アプローチ

上記課題に対する要求工学の分野としては、ゴール指向要求分析、トレーサビリティ技術などがあるが、現実

的には、単純に要求工学技術を当てはめれば、それですぐに問題が解決することはほとんどないだろう。

実際のソフトウェア構築の多くは人間系の作業によって成り立っている。また、その傾向は上流工程になればなるほど顕著になり、特に要求工学プロセスにおいては、開発する人間だけではなく、普段システム構築にはあまり携わらないユーザまでもを含めた人間の作業が主体となっている。よって、作業を実施している人間の負荷・コスト・能力を前提にせず、あるべき論での技術を適用しようとしても、結果として「使われない」「使えない」技術になってしまう。

すなわち、要求工学においては、対象としているプロセス内に、顧客を含めた人間的な作業を非常に多く含んでいるだけに、「要求工学」を必要としている人間の視点と「要求工学」の技術的・工学的視点とのギャップが大きく、そのギャップを埋める工夫が非常に重要となる。その意味で、第1節で述べた「ギャップ」の問題と、要求工学自身が本質的に抱えている課題は極めて相似的である。

要求工学を利用する側にとって要求工学を実際に役立つものにするためには、現場の課題に対して、適切に要求工学の技術を組み合わせつつ、現場で利用可能な形に変換する作業が必要となる。つまり、第2節で述べたように、要求工学プロセスに関連して各企業が抱えている課題の根本的な原因をミクロレベルまで分析した上で、改善すべきポイントを明確にし、それに対して適切な要

素技術を組み合わせるという作業が必要になる。

野村総合研究所においては、第2節で述べたような課題を解決するために、「システム機能レベルの要求を定義する際も含め、ユーザ視点を常に持つような要求構造とプロセスを適用することにより、『要求獲得・検証』をユーザの視点で実施できるようにすること」を基本的なアプローチとした。具体的には、我々は、ソフトウェアの仕様を、機能単位にまとめて記述するのではなく、ユーザの視点と機能の視点が交差する単位に記述するような要求の構造を適用することで、機能視点でソフトウェアの仕様を捉えられると同時に、ユーザ視点でも獲得・記述・検証できるようにした（図5）。

また上記を実現する手順としては、まず、プロジェクト計画の一部としてプロジェクトの性質に適した要求の構造やプロセスを計画するプロセス、次に計画の中で定義された要求の構造にしたがって要求を獲得・分析・検証するプロセス、さらに、獲得された要求に対する変更・関連性分析を実施する要求管理プロセスを定義した。また上記の考え方に基づいた、要求工程計画のガイドライン、テンプレート・ツールの作成を実施した（図6）。

上記プロジェクトを実施する中で特に気を使ったのが、いかに現場に負荷をかけない構造・プロセスにするかという点である。例えば、要求間の関係性を定義するトレーサビリティについても、当初は、ありとあらゆる関係要素間にトレースを張ることを想定していた。しかし、実際にトレースを張る作業は非常に負荷が高く、また、

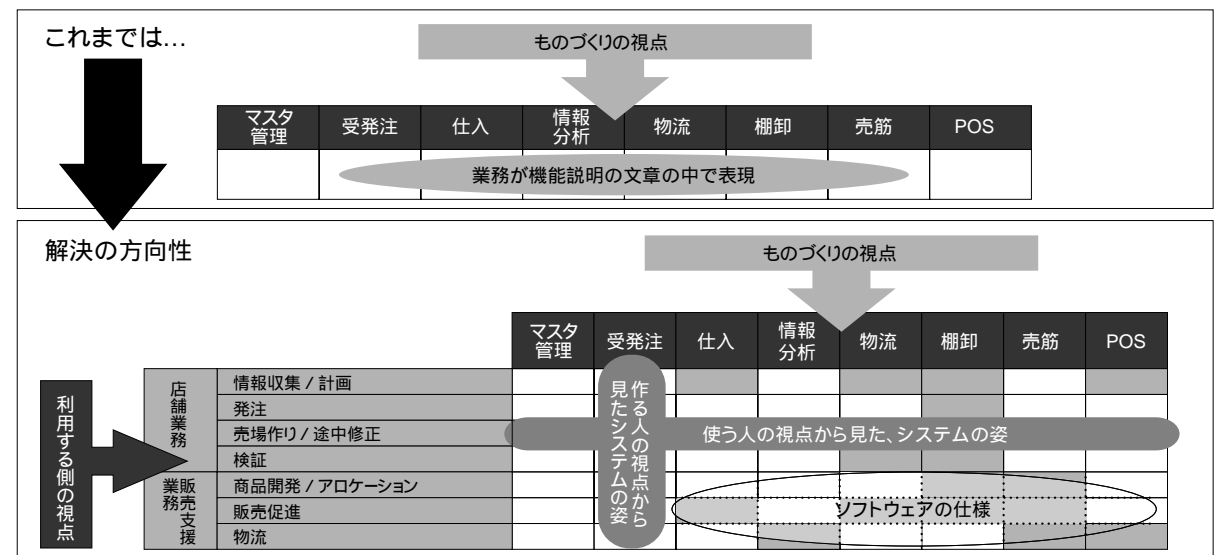


図5 野村総合研究所での要求構造のイメージ

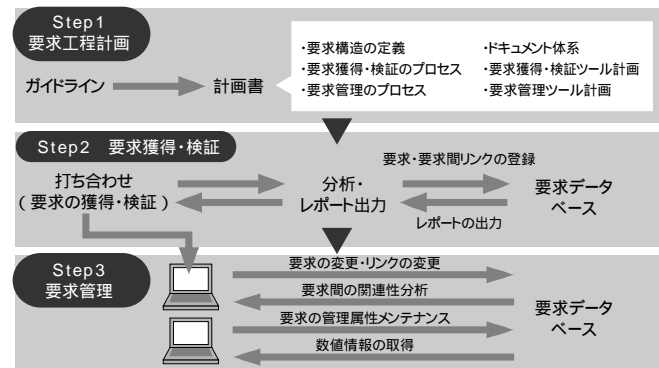


図6 要求工学のプロセス例

信頼性を維持することも困難であった。そこで、まずは上位レベルから業務視点で要求が整理できることを目指し、図5のマトリックスにある「利用する側の視点」と「ものづくりの視点」間のトレースが大枠としてとればよいとした。

また、トレースを、要求獲得・検証の作業の中で関係付けるトレース（構造トレース）と、要求を記述した後に網羅性の検証に利用するトレース（分析トレース）とを分けて考えた。また、わかりやすさ・管理しやすさを考え、構造トレースはすべて1:nでトレース可能となるように定義し、要求構造に対する理解が十分されなくても、作業中に間違いが生じ難いようにした。

結果として、粗粒度ではあるが要求間のトレースをとることにより、顧客・開発者ともに業務要求とソフトウェア仕様間の関連を理解できるアウトプットを作成し、いくつかのプロジェクトでは正確な要求の早期獲得・設計品質の向上を実現することができた。現在は、上記の考え方をベースに複数のプロジェクトに展開しながら、社内標準プロセスへの取込みを検討しているところである。

上記の要求工学プロセスの課題とそれに対する適用の事例は、所謂エンタプライズ系のアプリケーション開発の機能要求的側面に対するもので、必ずしも全ての企業・組織に当てはまるものではないだろう。しかし、他の企業が抱えている課題と共有する一般的な課題と捉えられる部分もあるであろう。

産業界と学术界で共有できる高度でかつ実践的な要求工学を追求するためには、要求工学を利用する側の課題と、それに対する要求工学の適用による解決事例、さらに、適用する際に利用した技術を含めた適用パターンを、産業界と学术界で共有・資産化する作業が核となる。さ

らに、それを他企業・組織に対しても横展開しながら、適用のパターンをノウハウとして蓄積していく。また、その過程で、学術面からそれぞれの適用パターンに採用される要求工学の要素技術を検証し、さらに発展させていく相乗関係のスキームを作っていくことが、要求工学を発展させていく上で要となるだろう。

4. 実践的な要求工学の実現に向けて

最後に、要求工学プロセスの適用の過程で筆者が感じた下記の2点について述べさせていただきながら、実践的な要求工学実現のための方向性を示したい。

(1) ユーザを視野に入れた要求工学を

要求が生み出される起点は、人と人との間の自然言語によるコミュニケーションである（この点については、[4]にて興味深い議論がされているので、ぜひ参照いただきたい）

要求工学プロセスとは、その自然言語によって行われるコミュニケーションから発生する定性的な要求をいかに形式的、かつ、体系的に獲得し、表現し、検証し、さらに、管理するかというプロセスである。よって今まで定性的にコミュニケーションされていた部分に、ある程度の形式性を取入れる必要がある。

もし、要求を獲得する起点となるユーザとのコミュニケーションが従来と変わらなければ、獲得後にその要求を形式的なものに変換しながら分析・検証しなければならない。その分作業負荷が大きくなり、プロセス改善の実現可能性が減ってしまう。そのため、ユーザに対して、適用するプロセスの考え方とその実現のためにユーザにして欲しいことをあらかじめ説明し、同意を受けることが望ましい。また、それをスムーズに実施するためには、ユーザに対して教育面でも工夫をする必要があるだろう。このように、ユーザとのコミュニケーション・プロセスにも、要求工学のスキームを入れることが、要求工学プロセスの出発点を制することであり、大きな成功のポイントである。

しかし逆説的ではあるが、ユーザの協力を得るためには、ユーザにとって表現・検証可能な形、つまり従来の

定性的なコミュニケーションになるべく近い形に要求を表現する必要があるだろう。特に現場に近いユーザに理解してもらうには、新業務や新システムをユーザが普段利用する「ビジネスの言葉」で「具体的に」「わかりやすく」表現する工夫が必要である。

ユーザの視点で表現可能・検証可能とするためには、ユーザが理解できることが前提であり、モデリング言語を利用する場合も、ユーザが理解可能であることが前提となるのはいうまでもない。関係するユーザの持つ技術的バックグラウンドを鑑み、また、要求獲得・表現・検証の作業負荷とのバランスを考えながら、教育的な工夫を実施し、時と場合に応じて適切なレベルの形式性を導入することが必要である。

(2) 「日本型」組織文化に根ざした要求工学を

日本においては、ユーザとベンダとの役割が曖昧なことが課題であるという考え方がある。一方で、両者の立場を超えてユーザとベンダとが、一体となって製品開発に取り組むことが、日本型企業の強みであるという考え方もある。

重要なのは、各企業が強みとして持っている組織文化を前提としながら、そこに、一定の形式・技術などの工夫を持ち込むことによって、その組織文化のよさをさらに生かす姿勢であろう。前述したように、要求工学プロセスの作業は人間系の作業が膨大であるがゆえに、特に組織文化が顕著に関わる部分である。その点、欧米特に米国の組織文化・企業文化に基づいた要求工学が必ずしも適用可能であるとは限らない。

移民国家である米国の、参加型組織形態を前提に個人（もしくはもしくは個々の会社）の役割を交換可能にし、その裏返しとして個々の役割を明文化することを重視してきた組織文化と、日本のように所属型組織形態を前提とし、その結果個々の役割は曖昧に定義され、役割以上の責任を果たす組織文化とは、適用すべきパターンは必ずしも同一のものにはならないだろう。

例えば、第2節で述べた打ち合わせ議事録量の推移は、ユーザとベンダ間ですり合わせながらボトムアップ的に次第に要求を決定していく日本型プロセスの現れであるという見方を仮にするならば、要求を前倒してトップダウン的に決定することは必ずしもあるべき姿でなく、一

定の体系の中で要求を決めていくことによって、要求が獲得される過程においても要求の整合性を保証する手段を提供することが要求工学プロセスを改善する1つの方向性になるであろう。

当然ながら、欧米型の技術やプロセスに表現されている原理・原則の多くは役に立つ普遍性を持っているし、その多くは日本にも適用可能であるだろう。しかしながら、特に人間系の作業が多く絡む要求工学の分野では、適用のパターンは、違うものになる可能性も考慮する必要がある。

オフショア開発などを通じて、国ごとにものの考え方、組織文化の違いが顕著にプロジェクトに影響してくることは多くの方が認識していることと思う。日本に適用可能な要求工学を目指すのであれば、日本的な組織に根付くものを追求していく姿勢を持ってほしい。

5. おわりに

要求工学をより実践的なものにしていくためには、産業界内のノウハウの共有と、それに対する学术界での研究成果からのフィードバックが不可欠である。そのためには、産業界と学术界との接点を作り、ノウハウを共有・蓄積するスキームが必要となってくる。そのスキームの構築こそ、SECに課せられたミッションであり、SECでこそ果たせる役割である。

そのようなスキームのもと、産業界の各社における現場の具体的な課題レベルから共有しながら、要求工学の適用、ノウハウの共有を進めることによって、日本の産業界にも適用可能な日本の要求工学が確立され、将来のプロジェクトの失敗を防ぐ端緒となることを期待したい。

参考文献

- [1] 斑目廣哉：『エンタプライズ系ソフトウェア・システム開発への取り組みについて』、<http://www.ipa.go.jp/software/sec/download/files/event/20041116/symposium.zip>
- [2] 山本修一郎：『連載要求工学 - 要求工学の概要 - 』、ビジネスコミュニケーション、10月号他、<http://www.bcm.co.jp/>
- [3] Jim Van Buren and David A. : Cook: Experiences in the Adoption of Requirements Engineering Technologies , Crosstalk , pp3-10 (Dec. 1998) , <http://www.stsc.hill.af.mil/CrossTalk>
- [4] Requirements Engineering Roundtable , SEI Interactive (March 1999) , <http://www.sei.cmu.edu/news-at-sei/features/1999/mar/Roundtable.mar99.pdf>

社団法人トロン協会

トロンプロジェクト

URL : <http://www.assoc.tron.org/>

社団法人 トロン協会 事務局

大橋 博

トロン (TRON: The Real-time Operating system Nucleus) は、理想的なコンピュータアーキテクチャの構築を目的として、1984年に東京大学の坂村健博士によって提案された新しいコンピュータOS仕様であり、産業界と大学の協力のもとで、新しいコンピュータの体系の実現を目指している。ここではトロンプロジェクトの中核の1つである社団法人トロン協会の活動を紹介する。

1 トロンプロジェクトの概要と活動経過

(1) 基本コンセプト

・どこでもコンピュータ

ユビキタス・コンピューティングが世界的に注目されてきているが、トロンは、発足当初から「どこでもコンピュータ環境」の構築を目的として始められたプロジェクトである。

また、「デジタルデバイド」が大きな社会問題となっており、コンピュータは誰でも使えるものでなければならぬとの観点から、プロジェクト開始直後から「イネーブルウェア (Enableware)」というコンセプトで、障害者対応も考えてきた。

さらに、どこでもコンピュータ環境におけるネットワークのセキュリティ保証が重要課題となっており、そのための標準セキュリティ基盤としての「eTRON」の構築を行っている。

他にも次の特徴を持つ。

・オープンアーキテクチャ

トロンプロジェクトの成果は公開され、このトロン仕様をもとに誰でも自由に製品を開発し市場に参入できる。

・弱い標準化

トロン仕様は、OS本体ではなくOSインタフェースを規定する。これを「弱い標準化」という。インタフェースは、階層的に定義されているのでそれぞれの階層を互いに別々に開発・実装することができる。1つのシステムでも、異なった層を別々の組織で開発できる。また同じ階層も自由競争の下で実現することができる。

(2) 活動の経過・実績

トロンプロジェクトは1984年に発足して以来、昨年で20年目を迎えた。

この間ITRON、BTRON、CTRONの各仕様書を世に出すとともに、トロンシンポジウムを筆頭に各種シンポジウム、セミナーを開催、トロンプロジェクトジャーナル等を刊行し、トロンの普及、啓蒙を行ってきた。なかでも、ITRONについては、携帯

表1 TRONの各プロジェクトの実績

プロジェクト	実績
ITRON	国内の組込み分野では、デファクト・スタンダード
BTRON	多文字OS「超漢字」として広く使われている
CTRON	電子交換機等に使用され、東西NTTの国際調達仕様
RTOS技術者教育	教育セミナーを全国各地で実施し、RTOS技術者の早期育成に寄与

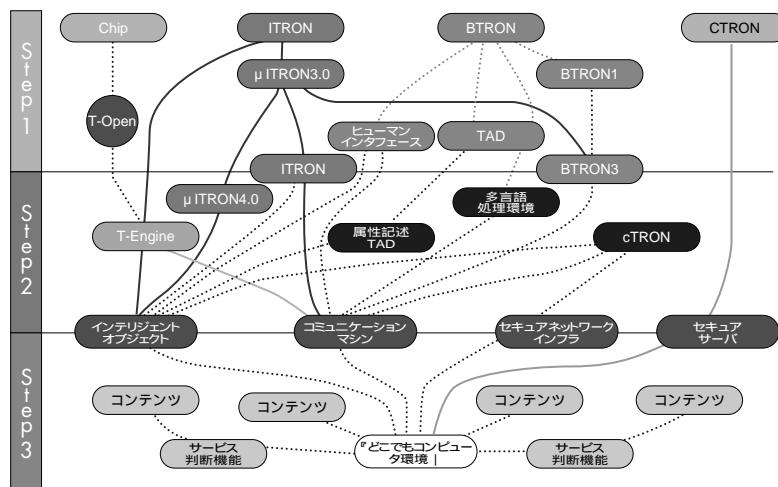


図1 トロンプロジェクトのロードマップ

電話をはじめデジタル家電分野で広く使用され、日本の組込み分野ではデファクト・スタンダードの地位を得た。トロンプロジェクトのロードマップは図1に示すとおりである。

2 トロンプロジェクトの今後の取り組み

(1) 課題認識

今後の課題は、T-Engineフォーラムとの連携強化によるITRONの一層の普及促進のための開発環境整備、JTRON・BTRONの普及促進、ユビキタスネットワーク社会の進展に対応したセキュアなeTRONの実証にある。

(2) 今後の取組み

トロンプロジェクト推進のための活動主体を以下の5つのグループに集約して活動を行う。各グループの活動については、研究開発指向を徹底させ、一層の活性化を図る。以下に各グループの活動内容を記す(図2)。

先端研究開発グループ

セキュアなデータキャリアチップ (eTRON) や、次世代ユビキタスコンピューティング環境 (HFDS) のインフラの構築に向けた、先進技術の研究開発、およびそれと関連する動向調査などを行う。

ITRON仕様検討グループ

T-Engineフォーラムとの連携を密にしT-Engineへの移行をスムーズに行うため、ガイドブックを作成した。今後は次の活動を行う。

・ITRONからT-Kernelへの移行に関する活動

・T-Engine Applianceに関する活動

・カーネルに関する技術活動に向けた検討

TRON多文字応用グループ

今迄の活動成果をもとにトロン・フォント・トレーサビリティ・システム (TFTS) を開発した。今後は電子政

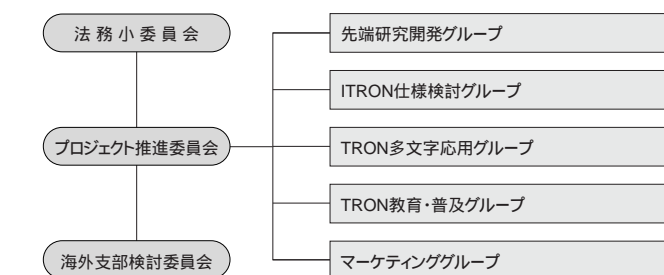


図2 社団法人トロン協会のプロジェクト推進体制

府をはじめとする地域情報システムや、電子ブックシステムなど、BTRON仕様OSの多文字応用を促進する活動を行う。

TRON教育・普及グループ

リアルタイムOS技術者の教育については、2004年は関西、九州、北海道を含め全国展開を行うとともに、入門者を対象としたWeb Learningも具体化してきた。今後は、実習を含む応用講座の拡充を図ることとし、Web Learning、入門講座、応用講座を3本柱に年間1,000人規模の技術者教育に本格的に取り組む(図3)。

マーケティンググループ

トロンプロジェクトの活動成果のマーケティング、プロモーション活動を行う。

その他

アジア地区をはじめとした海外普及活動に積極的に取り組む。

参考文献

- 坂村健: ユビキタスコンピュータ革命, 角川書店, 2002年6月
- 坂村健: 21世紀日本の情報戦略, 岩波書店, 2002年3月
- 坂村健: 情報文明の日本モデル - TRONが拓く次世代IT戦略, PHP研究所, 2001年10月
- 坂村健: ユビキタス TRONに出会う「どこでもコンピュータの時代へ」, NTT出版, 2004年10月
- 坂村健: 「未来社会におけるインフラストラクチャとしてのコンピュータとTRONプロジェクト」, 情報処理, Vol.35, No.10

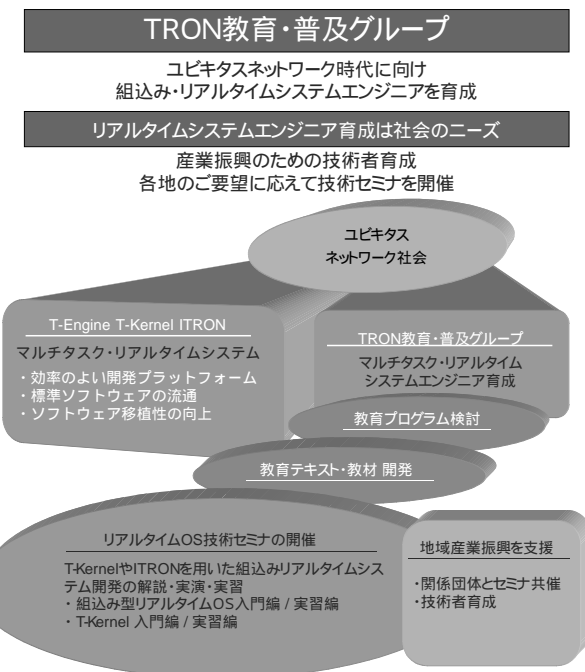


図3 TRON教育・普及グループ

北陸先端科学技術大学院大学 e-Societyプロジェクト

高信頼な組み込みソフトウェア設計技術の実用化を目指す

<http://kt-www.jaist.ac.jp/project/esociety/>

北陸先端科学技術大学院大学 情報科学研究科
教授 工学博士

片山 卓也

北陸先端科学技術大学院大学 情報科学研究科
産学連携客員教授 博士(情報科学)

岸 知二

北陸先端科学技術大学院大学 情報科学研究科
助手 博士(情報科学)

青木 利晃

北陸先端科学技術大学院大学 情報科学研究科
研究員 博士(情報科学)

岡崎 光隆

文部科学省のe-Societyプロジェクトの一環として、高信頼な組み込みソフトウェアの分析・設計を支援するツールや技術の研究開発が進められている。この活動の中心となっているのが、北陸先端科学技術大学院大学 片山研究室を中心としたグループである。このプロジェクトは最新のソフトウェア科学、ソフトウェア工学の成果を、実際の組み込みソフトウェア開発に適用するために、企業と密接なフォーメーションを組んで研究開発を進めている。本稿では、本研究開発の体制や研究環境について紹介する。

1 e-Societyプロジェクトとは

文部科学省リーディングプロジェクト基盤ソフトウェアの総合開発(e-Societyプロジェクト)では、これからのIT社会を支えるソフトウェア技術の開発を進めているが、その中の1つに「高信頼性組み込みソフトウェア構築技術」プロジェクトがある。本テーマでは高信頼性にかかわる3つの観点から研究開発を進めているが、高信頼な組み込みソフトウェアの分析・設計を支援するツールや技術の研究開発を担当しているのが、北陸先端大学・片山研究室を中心としたチームである。

2 北陸先端大学・片山研究室

正式名称は国立大学法人北陸先端科学技術大学院大学であるが、通常は北陸先端大学、あるいはJAIST (Japan Advanced Institute of Science and Technologyの頭字語) と呼ばれる。石川県にあり、東京から小松空港へ飛び、そこから空港連絡バスで、約30分のアクセスである。1990年設立で、現在は片山研究室のある情報科学研究科の他、知識科学研究科、材料科学研究科の3研究科体制である。

片山研究室では、一貫してソフトウェアの科学的側面についての研究を続けているが、e-Societyプロジェクトでは、形式的手法等を用いたソフトウェアの開発手法を組み込みソフトウェア開発に適用しようと取り組みを進めている。組み込みソフトウェアの規模や複雑さが増大する

中で、最新のソフトウェア技術の適用が急務であるからである。特に組み込みソフトウェアは、起こり得る様々な状況を想定し、どのような場合でも問題を引き起こさないように注意深く設計をする必要があるが、そうした組み込みソフトウェアの特性に、科学的手法は有効であると期待される。大学での長年の研究の成果が、本プロジェクトによって実用化へとつながっていきこうとしている。

なお、SECでは組み込みソフトウェアに関し、産業界のリアルな問題に取り組んでおり、本プロジェクトの活動にとっても大いに参考となる。SECと片山研究室は共同研究の計画を進めており、今後様々な成果が、産業界で効果的に活用されるようプロジェクトを進めていきたい。

3 プロジェクトの体制

e-Societyのプロジェクトでは、上に紹介した片山研究室メンバを中心に、図1に示す体制で活動している。

本プロジェクトの目的は、成果の実用化ということであり、産業界とのフォーメーションがキーとなる。そのため、大学、ツール提供企業、利用者企業との3者が密に協力しあう体制で進めている。

大学側：北陸先端科学技術大学院大学の片山卓也(教授) 岸知二(産学連携客員教授) 青木利晃(助手) 岡崎光隆(ポスドク) 博士課程学生3名、および国立情報学研究所の中島震(教授)のチームで、最新の科学的

法、工学的手法を組み込みソフトウェアにどう適用すべきか、その手法やツールの開発や設計を行うとともに、産業界との協調のもと、それらの評価や改良を進めている。

ツール提供企業：実開発経験や現場適用のノウハウに基づき、大学側と協調してツールのあり方を検討し、それを実際の開発に使える形に具現化を行っている。

利用者側企業：手法やツールを実際に適用・評価するための事例を提供するとともに、様々な評価を行っている。この評価は、大学側やツール提供企業にフィードバックされている。

上述した3者は週次、月次での頻繁な交流を行っており、単に大学側のアイデアを一方的に企業に持ち込むのではなく、企業側と一緒に問題を考え、アイデアを出し合うというスキームでプロジェクトが進められている。

なお、冒頭に述べたように「高信頼性組み込みソフトウェア構築技術」プロジェクトでは、3つの観点から検討を進めているが、他の2つのテーマを担当している早稲田大学中島達夫教授、京都大学湯浅太一教授の両グループとも定期的な技術・情報交流を続けており、平成17年度には、相互の成果の関連付けについて、より密な協調が開始する予定である。

こうした協力体制の下、プロジェクト2年目の現在、複数の企業の組み込み事例に対して、実際にモデル検査技術等の形式的手法を適用しながら研究が進行している。これらの活動を通して、実問題に科学的手法を適用する際の様々な問題を1つひとつ識別し、それに対する手法面、ツール支援面から解決策の検討を進めている。プロ

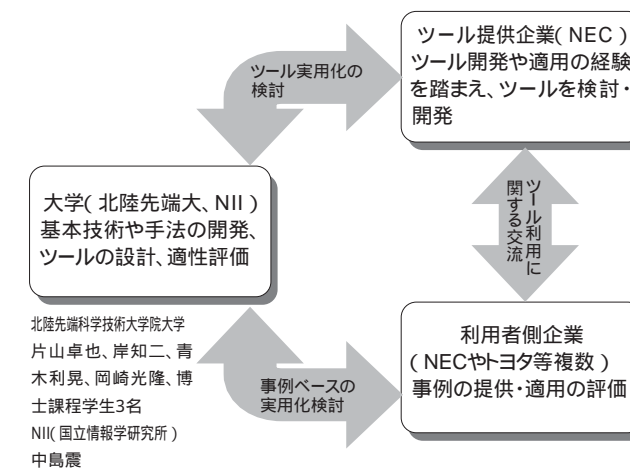


図1 プロジェクトの体制

ジェクト3年目には、支援ツールの実用バージョンをリリースする予定となっている。

4 研究環境

ところで、北陸先端大学の研究環境の特徴をいくつか紹介したい。

第一に、3研究科の大学院大学というコンパクトな体制の中で、教員、研究員、修士課程以上の学生が活発に研究活動を行っている点が指摘される。また新しい大学だけに、自由な雰囲気でのびのびと研究がなされている。

第二に、留学生や社会人の比率が高いことが挙げられる。様々な価値観、背景、経験を持った研究者がお互いに交流することで、刺激的な研究が進められている。ゼミでは日本語と英語が飛び交うこともまれではない。

第三に、石川県という地理的な特徴がある。北陸先端大学に来る人は、遠路田舎に来たという印象を持たれるし、実際そうなのだが、その適切な距離の存在を活用することにより、企業との協調作業と、集中した研究活動とを切り替えて行うことができる恵まれた環境ということができる。

以上紹介したように、ここでは最新の科学的、工学的手法の実用化に向けて、特徴のある、ダイナミックな研究活動が進められている。



図2 片山研を中心としたe-Societyメンバ



図3 研究室での活発な議論

慶應義塾大学 天野研究室

URL : <http://www.am.ics.keio.ac.jp/top/index-j.shtml>

慶應義塾大学理工学部 教授 工学博士

天野 英晴

慶應義塾大学理工学部天野研究室では、コンピュータを含むシステム全体の「アドバンスドアーキテクチャ」について並列処理、リコンフィギャラブルシステム、相互結合網をキーワードに広く研究している。

1 アドバンスドアーキテクチャとは？

コンピュータは、それだけで独立した巨大システムであり、このシステムをいかに構成するかを研究する「コンピュータアーキテクチャ」は、コンピュータサイエンスの重要分野の1つである。このアーキテクチャ分野の技術革新により、コンピュータは驚異的な発達を遂げ、既に多くの用途で満足すべき性能を備えるようになった。一方で、携帯電話、情報家電などコンピュータをその一部に含むITシステムが発達すると、コンピュータ単体の構造を研究するよりも、様々な種類の複数のプロセッサ、専用ハードウェア等を含むシステム全体をいかに高性能、低コスト、低消費電力で構成するかが重要になってきた。このため、コンピュータを含むシステム全体に「コンピュータアーキテクチャ」の概念を拡張する必要がある。我々、慶應義塾大学天野研究室は、複数のプロセッサ、専用ハードウェア等を含むシステム全体を構成する技術を「アドバンスドアーキテクチャ」と称して、これを理論から実践まで広く研究している。

2 研究プロジェクト紹介

我々の得意とする技術は、「並列処理」「リコンフィギャラブル」「相互結合網」である。年によって若干の違いはあるが、約30名の学生が、この技術に基づき様々なアドバンスドアーキテクチャの研究に取り組んでいる。博士課程の学生、在職ドクター、留学生も多く、スタッフはバラエティに富んでいる。この中の研究プロジェクトをいくつか紹介する。

(1) 動的リコンフィギャラブルシステム

情報家電、携帯端末等に用いられるSoC (System On-a-Chip) は、動画、静止画、暗号、復号化などに計算能力を必要とする一方、コストと消費電力に対する要求が厳しい。そこで、簡単な組み込みCPUと、計算能力を必要とする処理を専門に実行するハードウェアを組み合わせた構造をとる。しかし、これらのメディア処理の分野では新しい技術が次々に登場し、これに対応してその都度ハードウェアを設計し直すコストが大きくなっている。この専用ハードウェア部分を、書き換え可能なハードウェアで置き換えることができれば、単一のチップを様々な用途に使うことができ、チップ製造後に新しい技術を取り入れることもできる。しかし、現在の書き換え可能なハードウェアは柔軟性に優れている代わりに、柔軟性を実現するためのオーバーヘッドが大きく、面積効率の面で専用ハードウェアに遠く及ばない。この問題を解決する技術が、実行中にハードウェアの構造を短時間に入れ替えてしまう動的再構成である。この方法は同一の半導体面積を時分割で様々な処理に用いることができ、多くのケースで面積効率を10倍以上改善できる。我々の研究室は、この動的再構成技術を1992年に世界に先駆けて提案し、その実現化に向けて研究を重ねている。現在、NECエレクトロニクスと共同研究を行っており、同社の開発したDRP (Dynamically Reconfigurable Processor) をベースとして、アプリケーションの実装、アーキテクチャ的な解析と新たなアーキテクチャの提案、動的リコンフィギャラブルシステム用オンチップ結合網などの研究を幅広く行っている。

(2) バイオインフォマティクス用リコンフィギャラブルシステムの開発

書き換え可能なデバイス、特にSRAM型FPGAは、ここ10年間で急速に発達し、500万ゲートのランダムロジックに加え巨大なメモリ、CPU、DSP、乗算器等を組み込み、100MHzに及ぶ周波数で動く。リコンフィギャラブルシステムは、対象とする処理アルゴリズムを直接ハードウェア化し、書き換え可能なデバイス上で実行することによって柔軟性と性能の両者を実現する。動的リコンフィギャラブルシステムと違ってリコンフィギャラブルシステムは、ASIC化がコスト面で引き合わないニッチ分野での応用が進んでいる。しかし、単にアルゴリズムをハードウェア化するだけでは大きな効果は得られず、入力される問題に応じて、これに適した並列処理の構造を生成する技術が重要である。我々は、バイオインフォマティクス分野にこのリコンフィギャラブルシステムを用いる研究を、この分野の専門家と共同して行っている。図1に示すボードを開発し、細胞シミュレーション、細胞系譜作成システム、アミノ酸塩基配列照合等で、高性能PCの数十倍から数百倍の性能向上を達成している。このボードは他にも企業との共同でリコンフィギャラブルシステムを開発する際のベースとしても使っている。

(3) メモリスロット装着型ネットワークインタフェースの開発

安価で高性能のPCを多数接続して構築するPCクラスターは、現在スーパーコンピューティングの主演となっている。しかし、安価なPCを接続する場合はネットワークインタフェースを接続するPCIバスが転送性能のボトルネックになる。そこで、汎用のメモリスロットにネットワークインタフェースを接続してこのボトルネックを解消しようとするシステムがDIMMnet-2 (図2) である。このプロジェクトは、総務省の予算で実施され、農工大、東芝、横国大等との共同研究である。

(4) SECとの共同研究

現在の組み込み用のソフトウェアは、主としてメモリを節約するために様々な工夫が施されているため、ハードウェア化してリコンフィギャラブルシステム上で動作させる際に困難が大きい。天野研究室は、SECと共同で、画像、音声、暗号等の分野で用いられる各種組み込み用ソ

フトウェアをハードウェア化しやすい形に書き直す研究を2005年度よりスタートする。将来にわたり活用できる共通資産やベンチマークとなれば幸いである。

ここでは、誌面の関係で紹介できなかったが、これ以外にもPCクラスターやオンチップ内の相互結合網に関する理論的な研究、オンチップマルチプロセッサを想定したキャッシュ内蔵型スイッチ、ビジョンシステム、非同期システム等様々な研究が行われている。

3 天野研の特徴

我々の研究スタンスは以下の通りである。できるだけ実装してアプリケーションを動かして評価をとる。研究成果は国内外を問わず広く発表する。アプリケーションに密着し、企業との共同研究によって世の中にインパクトを与える研究を行う。現在、5社との間で共同研究を行っており、他の大学、研究機関との協力関係も広がっている。

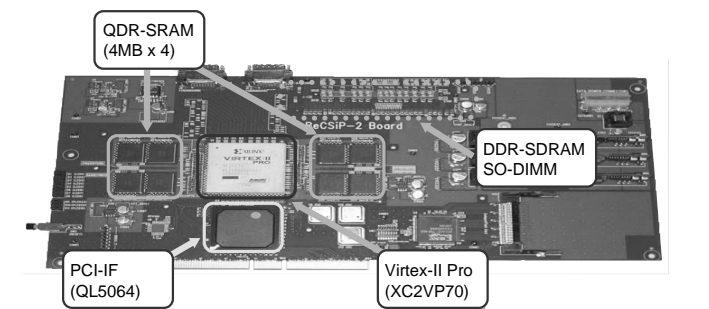


図1 バイオインフォマティクス用アクセラレータボードReCSiP

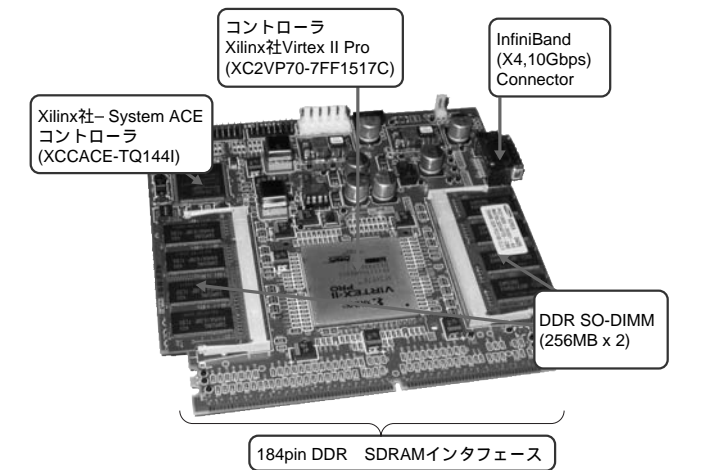


図2 DIMMnet-2用ネットワークインタフェースボード

昨年10月に設立されたソフトウェア・エンジニアリング・センター（SEC）も半年を迎え、第一弾の成果をWebにて順次公開致しました。さらに、エンタプライズ系プロジェクトでは、「ソフトウェア開発データ白書2005」
「経営者が参画する要求品質の確保」
「ITユーザとベンダのための定量的見積りの勧め」、組込み系プロジェクトでは、「組込みソフトウェア開発におけるプロジェクトマネジメント導入の勧め」
「組込みソフトウェア開発における品質向上の勧め ～コーディング編～」
「組込みスキル標準 ETSS概説書（2005年版）」等の出版が間近となっています。また、昨年の広報活動では、「Embedded Technology 2004」への出展・講演を始め、各企業や学会セミナーへの参加件数が15件と、多くの方々にSECの成果説明をする機会がありました。共同研究も東京大学経済学部ものづくり経営研究センターと「ものづくり競争力共同研究」について、ドイツ フラウンホーファー協会 IESEと「見積手法のフィジビリティに関する研究」についてそれぞれに着手し、また、今後の活動テーマが明確になり今後も継続して共同研究を行っていきます。

報告書（それぞれ、<http://www.ipa.go.jp/software/sec/download/>よりご覧頂けます）

「エンタプライズ系ソフトウェア開発力強化推進委員会活動報告(2005年4月)」

「組込みソフトウェア開発力強化推進委員会活動報告（2005年4月）」

執筆活動、取材記事

2004年	『情報処理』(情報処理学会 7月) 「解説 実証的ソフトウェア工学環境への取り組み」 執筆: SEC所長 鶴保証城、井上克郎(大阪大学)、松本健一(奈良先端科学技術大学院大学)、鳥居宏次(奈良先端科学技術大学院大学)
	『日刊工業新聞』(日刊工業新聞社 8月23日) 「日本のソフト産業強化～情報処理推進機構参与鶴保証城氏に聞く」 インタビュー: SEC所長 鶴保証城
	『経済産業新報』(経済産業新報社 9月5日) 「特集 わが国の競争力の要、組込みソフトウェア産業」 インタビュー: SEC研究員 門田浩、田丸喜一郎
	『経済産業新報』(経済産業新報社 10月5日 特別号) 「情報化月間記念座談会 ソフトウェア産業の今後の使命は?」 座談会: SEC所長 鶴保証城、JISA会長 佐藤雄二郎、METI情報処理振興課長 小林利典
	『経済産業新報』(経済産業新報社 11月15日) 「SEC設立記念 組込みソフトウェアにエンジニアリング思考を持ち込むSEC」 インタビュー: SEC所長 鶴保証城
	『経済産業新報』(経済産業新報社 11月15日) 「SEC設立記念座談会 組込みソフトウェア産業強化のために」 座談会: SEC研究員 門田浩、田丸喜一郎、平山雅之、渡辺登、METI情報処理振興課課長補佐 祝谷和宏
	『日経コンピュータ』(日経BP社 2004.12.13号) 「特集 日本のソフト開発力を取り戻せ」 インタビュー: SEC所長 鶴保証城
2005年	『COMPUTER & NETWORK LAN』(オーム社 12月号) 「特集 ユビキタス時代のソフトウェア・エンジニアリング」 インタビュー: SEC所長 鶴保証城 執筆: SEC研究員 石谷靖、新谷勝利、安田守
	『日経ものづくり』(日経BP社 1月号～連載) 「組込みソフト玉手箱」 執筆: SEC研究員 田丸喜一郎
	『情報処理』(情報処理学会 2月～6回連載) 「組込みソフト産業の実態と開発の課題」 執筆: 第1回 SECリサーチフェロー 大原茂之、第2回: SEC研究員 平山雅之
	『日経ITプロフェッショナルズ』(日経BP社 2月号) 「実践的な開発方法論の有無がIT業界の将来を左右する」 インタビュー: SEC所長 鶴保証城
	『日経コンピュータ』(日経BP社 3月7日号) 「コラム 組込スキル標準5月に登場(p61)」
	『COMPUTER & NETWORK LAN』(オーム社 4月号) 「シリーズ: 今、日本に必要なソフトウェア・エンジニアリング 他分野と比べたソフトウェア開発の諸問題」 執筆: SEC所長 鶴保証城
	『日経バイト』(日経BP社 4月号) 「組込の最前線」 執筆: SEC研究員 室修治
	『デザインウェーブマガジン』(CQ出版社 4月号から連載) 「スキル、ついてますか 組込みソフトウェア開発の人間学」 執筆: SEC研究員 渡辺登
	『日経コンピュータ』(日経BP社 4月18日号) 『Close Up 開発力向上に向けSECが本格始動』 取材協力

Web活動

2004年10月1日、IPAサイト内に、SECのサイトを立ち上げました。

<http://www.ipa.go.jp/software/sec/>

SECの組織や事業内容の紹介、講演資料等のダウンロード、出版物の購入ページ、イベント情報等があり、随時更新しています。

出版情報（書籍の監修）

『組込みソフトウェアレポート2005』 発売: 2004年11月	発行: 株式会社翔泳社 本体価格2,300円+税 解説: 組込みソフトウェア技術の代表的な企業、技術者の現場のインタビューと最新の技術動向を網羅。自動車、家電などの組込みシステムを図解でわかりやすく解説。組込み産業の人と技術を知る格好の一冊。
-------------------------------------	--

SEC成果物

エンタプライズ系 (白書)	『ソフトウェア開発データ白書2005 ～IT企業1000プロジェクトの定量データを徹底分析～』	発行: 日経BP社 本体価格2,286円+税 解説: 大手ソフトウェアベンダ15社のプロジェクトデータ1,000件以上を元に、ソフトウェア定量データベースを作成、統計処理、分析を行った結果を白書としてまとめた書籍。集計したデータの統計的プロフィール、規模・工期・工数・生産性・品質などについて、業種やシステム特性毎に分析した「業界初の分析データ」を示している。また、巻末にはデータ収集項目シートも添付されており、今後、データ収集・分析の強化を目指す企業には利用価値が高い。
エンタプライズ系 (小冊子)	『経営者が参画する要求品質の確保～超上流から攻めるIT化の動どころ～』	発行: 株式会社オーム社 本体価格500円+税 解説: 要求品質の確保のため動どころとして、経営者や企画・システム部門長が知っておくべき具体的な役割・成果や、要求から要件を明確にしていく上流過程を示す。また、上流工程の一部を、最重要工程ととらえ「超上流」と記述している。
エンタプライズ系 (小冊子)	『ITユーザとベンダのための定量的見積りの勧め～見積り精度を向上させるためのポイント～』	発行: 株式会社オーム社 本体価格300円+税 解説: ITソフトウェア開発時の見積り精度を向上させるためのポイントをまとめている。見積りの概念、重要性、実情と課題、ユーザとベンダそれぞれにとってのメリット、定量化の有効性、必要性を説明し、見積り時期毎に定量化すべき成果物やリスクと対象範囲等を特定することの重要性を示している。
組込み系 (小冊子)	『組込みソフトウェア開発におけるプロジェクトマネジメント導入の勧め』	発行: 株式会社翔泳社 本体価格300円+税 解説: 体系的なプロジェクトマネジメントの導入が遅れているといわれる組込みソフトウェア開発を対象に、プロジェクトマネジメントとは何か、導入することのメリットは何かを平易に解説した。あわせてプロジェクトマネジメントを実施する上で基本となる技術・知識についても紹介し導入への助けとしている。なかでも4章「プロジェクトマネジメントの心得」はマネジメントのエッセンスを要約したもので、組込みソフトウェア開発に関わっているすべての方にぜひ一読いただきたい章である。
組込み系 (小冊子)	『組込みソフトウェア開発における品質向上の勧め～コーディング編～』	発行: 株式会社翔泳社 本体価格300円+税 解説: ソースコードの良し悪しはソフトウェア品質の原点ともいえる。本書では、ソフトウェア品質の考え方とその原点としてのソースコード品質向上のための手段を紹介する。ソフトウェア品質の考え方としてはISO/IEC 9126の品質特性モデルを参考に組込みソフトウェア開発での考慮点を示す。また品質向上の手段としてはコーディング規約やそれを利用したソースコードレビューなどの方法を紹介する。
組込み系 (小冊子)	『組込みスキル標準 ETSS概説書(2005年版)』	発行: 株式会社翔泳社 本体価格300円+税 解説: 組込みスキル標準 (ETSS) の概要や活用方法等をわかりやすく解説した一冊。Version 1.0として公開される『スキル基準』を中心に、今回はDraft版として公開される『キャリア基準』や『組込みソフトウェア開発未経験者向けカリキュラム』も解説する。特に『スキル基準』において、組込みシステムに組込まれる技術要素をモデル化した『技術要素カテゴリ』については、開発現場で導入する際に必要となる考え方や例を紹介している。

ものづくり競争力共同研究

(東京大学経済学部ものづくり経営研究センター(MMRC))

東京大学21世紀COEものづくり経営研究センター (http://www.ut-mmrc.jp/) と SEC との連携は、2004年6月に事前準備を開始し、組み込みソフトウェアの製品への貢献度について共同研究を行うことで合意し、2004年8月から本格的に研究が始まった。

今や身の回りの多くの電化製品にソフトウェアが組み込まれているが、ソフトウェアが製品の価値あるいは価格にどの程度影響しているのか、また機能モジュール毎のソフトウェアの価値はどのように作り込まれるのかを明らかにすることが本共同研究のテーマである。

ソフトウェアが組み込まれた1つの製品が提供するいくつかの機能を考えたとき、ハードウェアで実現している機能もあれば、主にソフトウェアで実現している機能もある。もし、ソフトウェアで実現している機能が顧客にとって非常に魅力的で、その製品の特徴となるのであれば、その製品におけるソフトウェアの価値は高いと判断することができる。ソフトウェアの価値が高ければ高いほど、ソフトウェアの品質や開発効率が重要になり、

ソフトウェア・エンジニアリングが必要となる。そこで、いわゆるヘドニックプライスというアプローチを試みた。

製品の取引価格を決める要因としては、ハードウェアとソフトウェアの価値の他にも、製品や発売企業のブランド力、マーケティング力などがある。価格を決める上記要因の割合を特に製品機能についてより正確に推定するには製品の開発プロセスまで踏み込んだ分析が必要であることがわかった。今後は、開発プロセスにまで入り込んだ分析手法の確立を目指して共同研究を続ける。



MMRCにてソフトウェアの価値について説明する立本博文東京大学COE特任助手

IESE との共同研究

(ドイツ フラウンホーファー協会 IESE)

ドイツ フラウンホーファー協会 IESE (Institute of Experimental Software Engineering : 実験的ソフトウェア工学研究所) と2004年11月16日に共同研究覚書を交わし、2004年度は見積手法に関するテーマについて共同研究を行った(2004年11月~2005年3月)。具体的には、IESEで開発した2つの手法を日本の企業で実際に適用するものである。

それぞれOSR法(Optimized Set Reduction)とCoBRA法(Cost Estimation, Benchmark and Risk Management)と呼ばれるものであり、前者は大量の過去プロジェクトデータからデータマイニング手法を用いて、要因分析・モデル化を行うものである。後者は、プロジェクト経験豊富なPL等の専門家の知識を形式知化する手法である。

現在、ソフトウェア開発企業では、なかなかデータが集まらない例が多く、また、プロジェクトデータをまとめている場合であってもうまく分析結果が出ない例も多い。上記の手法は、そのような悩みを持った企業が自社に合った見積モデルを作成するニーズに応えられる可能性の

ある例として、実際の効果を試したものである。

1月下旬から2月上旬にかけては、IESEの研究員が2名来日し、企業でCoBRA法のモデル作成を行った。また、2社のプロジェクト実績データに対してOSR法で見積りモデルを作成した。両手法とも結果は良好で、現在さらなる評価と日本企業の特徴を生かした改善を引き続き研究しているところである。なお、2005年度前期は、2004年度の適用結果に基づき、2手法の改良と導入方法について、IESEと共同研究する。



SECにて手法の説明をするAdam Trendowicz IESE研究員

イベント報告(展示会参加、セミナー開催等)

開催日	タイトル	開催場所	主催	開催・参加内容	講演者・参加者(1)	後援タイトル・参加内容
2004年						
6月22日	組み込みソフトウェア開発力強化推進フォーラム	東京都港区・ラフォーレミュージアム六本木	経済産業省/IPA	IPAによるSEC活動概要の説明	後援者: 経済産業省 商務情報政策局 情報処理振興課 課長補佐 小林利典(経済産業省 商務情報政策局 情報処理振興課 課長) 組込みソフトウェア開発力強化推進委員会 委員長 門田浩 組込みソフトウェア開発力強化推進委員会 副委員長 田丸喜一郎 組込みソフトウェアエンジニアリング部会 部長 平山雅之 組込みソフトウェアスキル部会 部長 大原茂之 IPA 参与 鶴保証城 下堀友数(経済産業省 商務情報政策局 情報処理振興課 ソフトウェア課)	[開会の辞] 「我が国ソフトウェアの競争力強化に向けた政府の取組」 「組み込みソフトウェア開発力強化推進委員会の活動」 「組み込みソフトウェア産業実態調査の集計結果と分析」 「組み込みソフトウェアエンジニアリング部会の活動成果」 「組み込みソフトウェアスキル標準部会の活動成果」 「ソフトウェア・エンジニアリング・センター(SEC)とIPAの活動」 [閉会の辞]
9月9日~10日	IPAX 2004 Autumn	東京都千代田区・東京国際フォーラム	IPA	講演、パネルディスカッション等	組込みソフトウェア開発力強化推進委員会委員長 / IPA SEC 準備室研究員 門田浩 組込みソフトウェア開発力強化推進委員会副委員長 / IPA SEC 準備室研究員 田丸喜一郎	「コーディング作法とその効果」 「組み込みスキル標準の活用とキャリアデザイン」
10月1日	情報化月間	東京都港区・東京全日空ホテル	IPA	SEC設立発表会	SEC所長 鶴保証城 SECエンタプライズ系プロジェクトリーダー 安田守 SEC組込み系プロジェクトリーダー 門田浩	「ソフトウェア・エンジニアリング・センターの活動について」 「エンタプライズ系の活動について」 「組み込み系の活動について」
10月13日	ソフトウェアジャパン2004	東京都千代田区・明治大学 アカデミ-コモン	社団法人 情報処理学会	SEC鶴保証所長による講演	SEC所長 鶴保証城	ソフトウェア・エンジニアリング・センターの構想
10月14日~15日	組み込みソフトウェアシンポジウム2004	東京都江東区・日本科学未来館	社団法人 情報処理学会ソフトウェア工学研究会	SEC研究員、委員がセッションに参加	SEC研究員(キャッツ) 川井奈央	「状態モデルに基づいた組み込みプロトタイプ開発」
11月16日	調印式	IPA	SEC	調印式主催	[参加者]Dieter Rombach教授(ドイツ フラウンホーファー協会・IESE所長)、IPA理事長 藤原武平太、SEC所長 鶴保証城	ドイツ フラウンホーファー協会・IESEとの共同研究に関する契約締結、及び調印式
11月16日	ソフトウェア・エンジニアリング・センター設立記念講演会	東京都文京区・東京ドームホテル	SEC	講演会主催(2)	IPA理事長 藤原武平太 豊田正和(経済産業省 商務情報政策局 局長) 佐藤雄二郎(社団法人 情報サービス産業協会 会長) SEC所長 鶴保証城 鳥居宏次(奈良先端科学技術大学院大学 学長) Dieter Rombach教授(ドイツ フラウンホーファー協会・IESE所長) 坂目廣哉(富士通株式会社 取締役専務) 榎木好明(パナソニック モバイルコミュニケーションズ株式会社 取締役社長) 富永章(日本アイ・ビー・エム株式会社 取締役専務執行役員) 古川享(マイクロソフト株式会社 執行役 最高技術責任者 兼マイクロソフト コーポレートバイスプレジデント) 小林利典(経済産業省 商務情報政策局 情報処理振興課 課長)	
11月16日	ソフトウェア・エンジニアリング・センター設立記念レセプション	東京都文京区・東京ドームホテル	SEC	レセプション主催	出席者:関連団体・委員関連企業の招待者 出席者数:講演会388名・レセプション411名	
11月17日~19日	Embedded Technology 2004	神奈川県横浜市・パシフィコ横浜	社団法人 日本システムハウス協会(JASA)	スペシャルセッション主催、SEC研究員と委員がパネルディスカッションに参加。ブース展示参加、経済産業省ソフトウェア産業実態調査実施	小林利典(経済産業省 商務情報政策局 情報処理振興課 課長) SEC所長 鶴保証城 SEC委員 野中誠(東洋大学) SEC研究員 平山雅之 パネルディスカッション「ITSSとETSS」 [コーディネータ] 横田英史(日経バイト発行人 兼 編集長) [パネリスト] SEC委員 大原茂之(東海大学)、SEC委員 谷川流(トヨタ自動車)、SEC委員 二上貴夫(東陽テクニカ)、六反田隆(東芝情報システム)、吉田利夫(三菱電機)	特別講演「ソフトウェア開発力強化を推進する政府の取り組み」 「SECの活動とその成果」 「ソフトウェア特性ファイルによるソフトウェア構築プロセスの自動化」 「コーディング作法とインスペクション(現場で役立つ品質の作り込み)」 「スキルフレームワークと職種モデル」
12月1日~2日	第23回ソフトウェア生産における品質管理シンポジウム	東京都杉並区・日科技連東高円寺ビル	財団法人 日本科学技術連盟	SEC研究員、委員がセッション参加	SEC委員 二上貴夫(東陽テクニカ) SEC研究員 渡辺登	「組み込みソフトウェアの高信頼性保証について」 「組み込みソフトウェア開発スキルの体系的整理と活用」
12月3日	第1回国際ソフトウェア工学会議	韓国・ソウルCOEX(会議・展示センター)	MIC KIPA	SEC鶴保証所長による講演	SEC所長 鶴保証城	「日本の事例:信頼性の高いソフトウェア開発技術に向けて」 "Japan's Case: Highly reliable S/W development technology"
2005年						
1月24日~25日	ソフトウェアテストシンポジウム2005	東京都港区・東京コンファレンスセンター品川	ソフトウェアテストシンポジウム実行委員会(IPA協賛)	SEC研究員、委員がパネルディスカッションに参加	[パネリスト]SEC委員 三橋二彩子(NECエレクトロニクス) [パネリスト]SEC研究員 川井奈央、SEC委員 古山寿樹(松下電器)、SEC委員 二上貴夫(東陽テクニカ)、平林俊一(富士通)	ミニパネルディスカッション:コーディング作法のすすめ
1月27日	第4回リテラリティソフトウェアワークショップ	茨城県つくば市・宇宙航空研究開発機構 筑波宇宙センター	独立行政法人 宇宙航空研究開発機構	SEC鶴保証所長による講演	SEC所長 鶴保証城	「ものづくり」としての高品質なソフトウェア開発を目指して」
2月3日~4日	Developers Summit 2005	東京都渋谷区・青山ダイヤモンドホール	株式会社 翔泳社	SEC研究員がセッション参加	SEC研究員 渡辺登 SEC研究員 室修治	「組み込みスキル標準の開発と活用」 「組み込み開発におけるプロジェクトマネジメントの動向」
2月18日	アジャイルプロセス協議会 第4回定例セミナー	東京都港区・株式会社日システムアンドサービス カンパリスルームA	アジャイルプロセス協議会	SEC鶴保証所長による講演	SEC所長 鶴保証城	「ものづくり」としての高品質なソフトウェア開発を目指して」
2月22日	JASA組み込みソフトウェアフォーラムin名古屋	愛知県名古屋市中区・名古屋商工会議所	社団法人 日本システムハウス協会(JASA) 中部支部	SECリサーチフェロー、研究員がセッション参加	SECリサーチフェロー 野中誠(東洋大学) SEC研究員 関口正 SEC研究員 大野克巳	基調講演「プロジェクト/ソフトウェア特性プロファイルと組み込みシステム開発」 「スキル標準で描く高品質ソフトウェアを作る技術者像」 「高品質実装のためのコーディング作法」
3月2日~4日	情報処理学会 第67回全国大会	東京都調布市・電気通信大学	社団法人 情報処理学会	SECリサーチフェロー、研究員、委員がセッション参加 パネルディスカッション	SEC委員 高田広章(名古屋大学) SEC研究員 田丸喜一郎 [コーディネータ]SECリサーチフェロー 大原茂之(東洋大学) [パネリスト]小林利典(経済産業省)、SEC委員 清尾克彦(三菱電機)、SEC委員 高田広章(名古屋大学)、SEC委員 西康晴(電気通信大学)、SEC研究員 渡辺登	「組み込みシステム技術動向」 「組み込みソフトウェア産業の実態 - 2005年版技術者個人向け調査速報 -」 「組み込みシステム技術者の将来像と育成戦略」

1 所属は、イベント当時のものです。団体名、企業名等の表現は一般的な呼称を用いました。
2 講演内容については、SECサイト内の「イベント情報」でご確認ください。

BOOK REVIEW

技術革新と経済発展 非線形ダイナミズムの解明

弘岡正明 著

ISBN : 4-532-13242-8 日本経済新聞社刊
A5判・372頁・定価：5460円(税込) 2003年6月刊

人生を包む視野を得る

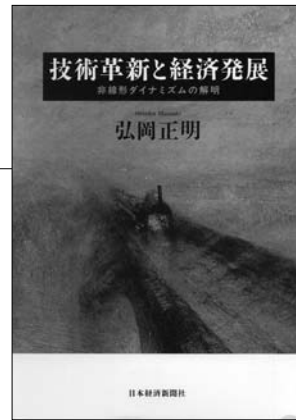
「経済学では取り扱われなかった科学技術の発展過程のダイナミズムを明らかにする」と帯封で謳う大著である。読者の視点を産業革命以降の全史を鳥瞰するスケールに広げてくれる。評者は特に次の3つの点で覚醒させられた。

1点目は、これまでの技術発展が、横軸を時間軸とするロジスティック曲線を描くという歴史的事実である。この事実を丹念な歴史検証で実証する著者の努力には敬服する。

2番目は、このロジスティック曲線が、発明に相当する技術軌道、実用化に相当する開発軌道、そして、ビジネスに相当する普及軌道の3本、平行移動の形で存在するというのである。読者は自分の挑戦領域についてこの軌道上の位置を考えると有用である。同じ技術テーマでも、基礎研究なのか、実

用化研究なのか、もはやビジネスレベルなのか、そしてその間隔は何年なのか、常に悩まされる課題だ。SECの挑戦するソフトウェア工学の領域でもこの課題はぴったり当てはまる。

3点目は、コンドラチェフの景気波動への着目である。この約57年周期の波動は産業革命以来4波観測され、目下その大底から第5波の立ち上げを迎えようとしている。この波を自分の人生と重ねて考えると一気に視界が開ける。人生50年時代では、1波しか体験できなかったが、現在は幸せにも2波体験できる。是非、第5波の立ち上げに、その牽引力の中心で立ち合いたいという気持ちになる。(神谷芳樹)



UML 動的モデルによる組み込み開発 - 分析・設計・実装・テスト -

渡辺政彦・飯田周作・石田哲史・山本修二・浅利康二 共著

ISBN : 4-274-06513-8 オーム社刊
B5変形判・392頁 定価：3,990円(税込) 2003年4月刊

モデルっていいかも

『モデル』はとても抽象的であり、実際に動くモノしか信用しない(できない)自分には苦手な話題である。

組み込みソフトウェア技術者の初級者教育として行われてきたUMLロボコン(今年からETロボコン)では、自律でライントレースし走行するモデルの評価と、実機でのタイムトライアルが行われる。昨年の結果では、優秀なモデルとタイムトライアルの結果に相関関係はなかった。優秀なモデルであっても、実際に速く走るわけではない。

しかし、この書籍で紹介されている事例と解説を読んでいると考え方が変わってくる。上流工程で仕様書・設計書をキチンと書き、有識者がレビューするのが昔からのセオリーである。ここでモデルという考え方を導入し、モデル検証まで含めて実施すれば、より品質を高められるはずである。システム間、

タスク間などのハンドシェイクや状態遷移をモデルベースで検証できれば、レビューでは信頼性や保守性など別の観点に注力することもできる。機械にできることは機械に任せ、人間は人間にしかできないことに注力したい。このモデル検証を普及させるには、この書籍にあるような組み込み向けの具体的なサンプルや成功事例などを増やす必要があるだろう。

ちなみに、UMLロボコンにおけるモデル評価は、高速に走ることに限らず、問題領域、責務分割、状態定義などモデリング技術が評価対象である。よって、「いいモデル」=「早いシステム」になるとは限らないのであった。なるほど...(渡辺登)



ソフトウェア・エンジニアリング関連 イベントカレンダー

作成：SEC journal 編集委員会

開催時期	開催日	イベント名	主催	開催場所	URL
5月	18日(水)~20日(金)	IPAX2005	IPA	東京都江東区・東京ビックサイト	http://www.ipa.go.jp/
	18日(水)	にいがた産業創造機構セミナー	財団法人 にいがた産業創造機構	新潟県新潟市・ユニソンプラザ	http://www.nico.or.jp/
	18日(水)	NPO法人人間中心設計推進機構 設立記念フォーラム	NPO法人 人間中心設計推進機構	東京都新宿区・早稲田大学 井深大記念ホール	http://www.hcdnet.org/050518_start.html
6月	8日(水)~10日(金)	ソフトウェアシンポジウム2005	ソフトウェア技術者協会	富山県富山市・富山国際会議場 http://www.ticc.co.jp/	http://ss2005.jaist.ac.jp/
	17日(金)	情報処理学会連続セミナー2005(第1回) 「組み込みソフト基礎と各社の状況」	社団法人 情報処理学会	東京都千代田区・東京電機大学 神田キャンパス7号館1F 丹羽ホール	http://www.ipsj.or.jp/
	20日(月)~21日(火)	SEC Forum 2005	SEC	東京都千代田区・経団連ホール	http://www.ipa.go.jp/software/sec/
	29日(水)~7月1日(金)	ESEC	リードエグジビジョンジャパン株式会社	東京都江東区・東京ビックサイト	http://web.reedexpo.co.jp/ESEC/
7月	29日(水)~7月1日(金)	SODEC	リードエグジビジョンジャパン株式会社	東京都江東区・東京ビックサイト	http://web.reedexpo.co.jp/SODEC/
	2日(土)~3日(日)	ETソフトウェアデザインロボットコンテスト	日本システムハウス協会(JASA)	東京都江東区・株式会社内田洋行潮見ビル	http://with.esm.co.jp/etrobo/
	8日(金)	情報処理学会連続セミナー2005(第2回) 「組み込み用OSの基礎と応用」	社団法人 情報処理学会	東京都千代田区・東京電機大学 神田キャンパス7号館1F 丹羽ホール	http://www.ipsj.or.jp/
8月	25日(月)~26日(火)	SPES2005	社団法人 情報サービス産業協会(JISA)	東京都港区・日本科学未来館	http://www.jisa.or.jp/seminar/spes2005paper.pdf
	8月25日(木)~26日(金)	SWEST(Summer Workshop on Embedded System Technologies)	組み込みシステム技術に関するサマータワーワークショップ 実行委員会	静岡県浜松市・遠鉄ホテルエンパイア	http://www.ertl.jp/SWEST/
9月	8日(木)	組み込みソフトウェア・マネジメント・フォーラム	翔泳社	東京都港区・泉ガーデンギャラリー	http://www.shoehisha.co.jp/
	14日(水)	情報処理学会連続セミナー2005(第3回) 「組み込み用LSI」	社団法人 情報処理学会	東京都千代田区・東京電機大学 神田キャンパス7号館1F 丹羽ホール	http://www.ipsj.or.jp/
10月	3日(月)	情報化月間	経済産業省	東京都港区・全日空ホテル	http://www.ipa.go.jp/
	7日(金)	情報処理学会連続セミナー2005(第4回) 「組み込みソフト開発手法・検証・ツール」	社団法人 情報処理学会	東京都千代田区・東京電機大学 神田キャンパス7号館1F 丹羽ホール	http://www.ipsj.or.jp/
	12日(水)~14日(金)	SEPG japan 2005	日本SPIコンソーシアム(JASPIC)	東京都港区・東京コンファレンスセンター品川	http://www.jasa.or.jp/
	17日(月)~19日(水)	組み込みソフトウェアシンポジウム2005	社団法人 情報処理学会 ソフトウェア工学研究会	東京都港区・日本科学未来館	http://www.ipsj.or.jp/
11月	24日(月)	SECソフトウェア・エンジニアリング・コンファレンス	IPA/SEC	東京都千代田区・経団連会館	http://www.ipa.go.jp/software/sec/
	16日(水)~18日(金)	Embedded Technology 2005	社団法人 日本システムハウス協会(JASA)(IPA協賛予定)	神奈川県横浜市・パシフィコ横浜	http://www.jasa.or.jp/
	25日(金)	情報処理学会連続セミナー2005(第5回) 「組み込みソフト開発事例(組み込みOS系)」	社団法人 情報処理学会	東京都千代田区・東京電機大学 神田キャンパス7号館1F 丹羽ホール	http://www.ipsj.or.jp/
	28日(月)	情報処理学会連続セミナー2005(第6回) 「組み込みソフト開発事例(コピキタス系)」	社団法人 情報処理学会	東京都千代田区・東京電機大学 神田キャンパス7号館1F 丹羽ホール	http://www.ipsj.or.jp/

上記は変更される場合があります。参加の際に必要な詳細事項は主催者にお問合せをお願いします。

SEC journal 第2号の発行が無事出来ました。今回も非常に役に立つ論文・技術解説を執筆いただいた方々に感謝いたします。目指すところは、「SEC journalの執筆依頼が来た」と執筆者が会社や大学内で自慢できるようになることです。海外レポート(30頁参照)は、鶴保所長・石谷研究員・新谷研究員の3名が強行スケジュールで、海外の4機関と共同研究の有り方について討論してきた報告で、SECの活動成果が世界で利用される第一歩です(言いすぎです)。また、今号は、2004年度の節目の時期にあたるので、SECの活動や成果物を振り返ってみました。SECは発足直前よりほぼ毎月のイベント参加によりSECの活動目的や解決すべき課題のご説明を行ってまいりました(61頁SEC成果報告参照)。エンタプライズ系委員78名、組込み系委員92名により討議された内容をSECで、5種類の小冊子・白書として出版することができました。30名のSEC研究員でこなしてきたと自慢したくなる気持ちと、SECの力不足を、委員の方々の力(中には一回の委員会で10:00から21:00と長時間の討議)により出版にこぎつけたと、深く感謝すると共に、まだまだ頑張らなければの気持ちです。今後もイベントに積極的に参加し(63頁イベントカレンダー参照)、加えて、Webをよりいっそう強化し、メールマガジンの配信も計画しておりますので、これからの活動にご注目ください。

なお、数々のイベントで「ソフトウェア・エンジニアリング・センター」を「SEC」と略して表現させていただいているので、今後はSEC journalにおいても原則「SEC」と表記させていただきます。

また、本Journalに対してのご意見もお待ちしております。<ご意見用メールアドレス: sec-journal@ipa.go.jp> (ヒゲ)

SEC journal 編集委員会

編集委員長

猪狩 秀夫(ソフトウェア・エンジニアリング・センター 組込み系プロジェクト)

編集委員

赤田 眞弓

石谷 靖

川井 奈央

関口 正

田丸喜一郎

樋口 登

神谷 芳樹

門田 浩

安田 守

渡辺 登



SEC journal 第1巻第2号(通巻2号) 2005年4月25日発行

独立行政法人 情報処理推進機構 2005

編集兼発行人 〒113-6591 東京都文京区本駒込2-28-8 文京グリーンコート センターオフィス16階
独立行政法人 情報処理推進機構 ソフトウェア・エンジニアリング・センター 所長 鶴保 征城
Tel.03-5978-7543 Fax.03-5978-7517
<http://www.ipa.go.jp/software/sec/>

編集・制作 〒101-8460 東京都千代田区神田錦町3-1 株式会社オーム社 Tel 03-3233-0641

本誌は、「著作権法」によって、著作権等の権利が保護されている著作物です。
本誌に掲載されている会社名・製品名は、一般に各社の商標または登録商標です。

SEC Forum 2005

SECは、6月20日、21日の2日間にわたり、
2004年度下期の活動と成果を報告する
「SEC Forum 2005」を開催いたします。

6月20日(月) 10:00~17:00

講演

「2004年度下期SEC活動と成果について」
SEC所長 鶴保征城

基調講演

「ユビキタス時代のソフトウェア(仮)」
東京大学大学院 情報学環・学際情報学府
教授 坂村健

活動報告「エンタプライズ系プロジェクト総括」

定量データ分析
「1000プロジェクトの分析から見る
ソフトウェア開発実態」
「ユーザ企業の実績データ分析から見る
ソフトウェア開発実態」
見積り手法
「プロジェクト成功のための
先進的見積り手法」
開発プロセス共有化
「『超上流』でのユーザ・ベンダの役割分担」

会場:経団連ホール(地下鉄大手町駅A1またはC1出口 徒歩1分) 入場無料。関係各位には追ってご案内状をお送りいたします。

6月21日(火) 10:00~17:00

特別講演

「2005年度 SEC活動に期待すること」
経済産業省 商務情報政策局 情報処理振興課
課長 小林利典

調査報告

「組込みソフトウェア産業実態調査報告」
SEC 組込み系プロジェクト サブリーダ 田丸喜一郎

活動報告「組込み系プロジェクト総括」

組込みスキル標準
「ETSS :
人材育成・活用のための組込みスキル標準」
コーディング作法
「組込みソフトウェア品質向上のための
コーディング作法」
プロジェクトマネジメント
「組込みプロジェクトを成功させる開発プ
ロセスとプロジェクトマネジメント」

「SEC journal」バックナンバーのご案内

No.1 創刊記念号

SEC journal創刊記念招待論文「ソフトウェア開発見積り式の汎用化の提案」
「最新ソフトウェア技術による高信頼組込みソフトウェアの開発」
/ 技術解説「エンタプライズ系ソフトウェア開発の課題」
「組込みソフトウェアスキル標準」
「組込みソフトウェア・エンジニアリング」/ 所長対談
H.D.Rombach教授 / 組織紹介「ドイツ・フラウンホーファ協会 IESE」
「EASEプロジェクト」
「東大ものづくり経営センター」
定価1,470円(税込)

バックナンバーは、SECサイト内「SEC journal」のページ
(<http://www.ipa.go.jp/software/sec/download/>)よりご注文いただけます。

